



**STUDENTS MANAGEMENT SYSTEM**  
**For**  
**SRI LANKA INSTITUTE of ARCHITECTS**

**P.M.N. SAMPATHPRIYA**

**BIT Registration Number: R020168**

**Index Number: 0201685**

**Supervisor: Mr. Fayaz Thahir**

**November 2017**



**This dissertation is submitted in partial fulfilment of the requirement of the  
Degree of Bachelor of Information Technology (External) of the  
University of Colombo School of Computing**

# DECLARATION

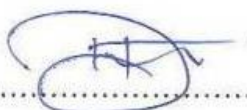
I certify that this dissertation does not incorporate, without acknowledgement, any material previously submitted for a degree or diploma in any university and to the best of my knowledge and belief, it does not contain any material previously published or written by another person or myself except where due reference is made in the text. I also hereby give consent for my dissertation, if accepted, to be made available for photocopying and for interlibrary loans, and for the title and abstract to be made available to outside organizations.

  
.....

Date: 05/11/2017

P.M.N. Sampathpriya


Countersigned by:

  
.....

Date: 05/11/2017

Mr. Fayaz Thahir

(Name of Supervisor)

  
.....

Date: 05/11/2017

Mr. Veranjan Kurukulasuriya

(Name of Advisor)

# ABSTRACT

The Sri Lanka Institute of Architects (SLIA) is the professional institution incorporated by an Act of Parliament empowered with the authority to regulate the architectural profession and education in Sri Lanka. The Board of Architectural Education (BAE) is the SLIA arm that manages the whole spectrum of activities related to education. In this process the BAE conducts lectures and examinations at the professional level for around 100 prospective architects each year. With the student-numbers increasing at a high rate, the BAE wishes to implement a centralized system to manage and monitor the students' academic and practical performances over several years.

After analyzing the current system and the processes followed at BAE, it was identified that the processes in place are handled manually using cumbersome paperwork. Thus it is very difficult to maintain proper records on students' admissions, attendance, payments, examination records and performing relevant activities in more efficient manner. A proposal was presented to the client by describing the possibility of implementing a web based Students Management System which will help to resolve the problems of the current system. The scope of the proposed system has been outlined with several modules such as students' admission process handling module, payments handling and notification module, attendance handling module, examination records and course records maintaining module and reports generating module.

The Rational Unified Process (RUP) was identified as the most suitable software development methodology for developing the Students Management System. XAMPP, NetBeans IDE, MySQL Workbench, HTML, CSS, StarUML and Bootstrap admin templates have been used as the tools and technologies in the design and implementation of the solution.

After completing a successful software development process, the implemented system was installed at the client's location for user acceptance testing and facilitating user trainings. The feedback received from the client is really positive and they anticipate the new system would immensely help for streamlining the organizational procedures progressively.

# ACKNOWLEDGEMENTS

I would like to express my sincere gratitude for the following people who immensely helped me to successfully complete my software development project work of the BIT final year. The Coordinators of BIT degree programme and the academic staff of University of Colombo School of Computing (UCSC) should be thankfully mentioned for creating a great path for under graduate students in the IT industry.

I would like to express my deepest appreciation to my project supervisor Mr. Fayaz Thahir for providing technical guidance throughout the activities of my project and concede that without his technical advices; the completion of the project wouldn't have been possible.

My heartfelt gratefulness should go to Mr. Veranjan Kurukulasuriya who is the Senior Vice President of Sri Lanka Institute of Architects and who is also my Advisor of the project for offering me unconditional help to achieve a great milestone of my life.

I would also like to thankfully mention the staff members of Board of Architectural Education of Sri Lanka Institute of Architects for helping me throughout the project with their valuable time and effort.

# TABLE OF CONTENTS

DECLARATION .....	ii
ABSTRACT .....	iii
ACKNOWLEDGEMENTS .....	iv
TABLE OF CONTENTS.....	v
LIST OF FIGURES .....	ix
LIST OF TABLES .....	xi
LIST OF ACRONYMS .....	xii
CHAPTER 01: INTRODUCTION .....	1
1.1 Introduction .....	1
1.2 Motivation of the Project.....	2
1.3 Objectives of the System.....	2
1.4 Scope of the Proposed Project.....	3
1.5 Structure of the Dissertation.....	5
CHAPTER 02: ANALYSIS .....	7
2.1 Introduction .....	7
2.2 Requirements Gathering Techniques .....	7
2.3 Analyzing the Current System and Processes .....	8
2.3.1 Students' Admission Process.....	8
2.3.2 Students' Payments Handling Process.....	8
2.3.3 Students' Attendance Handling Process .....	9
2.3.4 Examinations Handling Process .....	9
2.3.5 BAE Course Records Handling Process .....	9
2.4 Existing Similar Systems Analysis .....	10
2.4.1 Fedena School Management System .....	10
2.4.2 FeKara School Management System .....	12
2.4.3 mySkoolapp School Management Software Mobile App .....	13
2.5 Functional Requirements of the System.....	14
2.5.1 Students' Admission Process Handling Module.....	14

2.5.2	Students' Payments Handling and Notification Module .....	15
2.5.3	Students' Attendance Handling and Notification Module.....	15
2.5.4	Students' Examination Records Handling Module.....	16
2.5.5	BAE Course Records Handling Module.....	16
2.5.6	Staff Information Handling Module .....	16
2.5.7	Reports Generating Module.....	17
2.5.8	Blog and Students Discussion Forum .....	17
2.6	Non-functional Requirements of the System .....	17
2.6.1	Accuracy .....	18
2.6.2	Security .....	18
2.6.3	Reliability.....	18
2.6.4	User-friendliness .....	18
2.6.5	Maintainability.....	18
2.7	Overview of Software Process Models .....	18
2.8	Selected Methodology for the Proposed System.....	19
CHAPTER 03: DESIGN OF SOLUTION .....		21
3.1	Introduction .....	21
3.2	Alternative Solutions for the Proposed System.....	21
3.2.1	Purchasing a Commercial Product from a Software Vendor.....	21
3.2.2	Implementing the System as a Stand-alone Application .....	22
3.3	Implementing the System as a Web Based Application .....	22
3.4	Object-Oriented Systems Analysis and Design .....	23
3.4.1	High Level Use Case Diagram for the Students Management System .....	24
3.4.2	Class Diagram for the Students Management System.....	25
3.4.3	Use Case Diagram for the Students Registration Module .....	26
3.4.4	Use Case Diagram for the Fees Management Module .....	26
3.5	Database Design for the System.....	27
3.6	User Interface Design for the System .....	28
3.6.1	User Login Interface .....	29
3.6.2	Course Fee Payment Interface .....	29
3.6.3	Exam Results Upload Interface .....	30

CHAPTER 04: IMPLEMENTATION .....	31
4.1 Introduction .....	31
4.2 Implementation Environment.....	31
4.3 System Development Tools and Technologies .....	32
4.3.1 NetBeans IDE 8.2 .....	32
4.3.2 MySQL Workbench 6.3.....	32
4.3.3 XAMPP Server v3.2.1 .....	32
4.3.4 StarUML 5.0.2 .....	32
4.4 Major Code Structure .....	32
4.4.1 Database Connection Session .....	32
4.4.2 Success and Error Messages .....	33
4.4.3 User Login Handling Code .....	33
4.4.4 Logout Message Box .....	34
4.4.5 Insert, Update and Inactivate Code Segments .....	35
4.5 Module Structure of the System.....	36
4.6 Re-Usable Existing Components .....	37
CHAPTER 05: EVALUATION .....	39
5.1 Introduction .....	39
5.2 Software Testing vs Software Evaluation .....	39
5.2.1 Software Testing Process.....	39
5.2.2 Software Evaluation.....	41
5.3 System Test Plan .....	41
5.4 System Test Cases .....	43
5.4.1 Test Cases for Students Registration Module .....	43
5.5 User Acceptance Testing.....	45
CHAPTER 06: CONCLUSION .....	46
6.1 Introduction .....	46
6.2 Critical Assessment of the Project .....	46
6.3 Lessons Learnt During the Course of the Project .....	47
6.4 Future Enhancements .....	48
REFERENCES .....	49

Appendix A - System Documentation .....	51
Appendix B - Design Documentation.....	53
Appendix C - User Documentation.....	57
Appendix D - Management Reports .....	73
Appendix E - Test Results .....	78
Appendix F - Code Listing .....	88
Appendix G - Client Certificate .....	94
GLOSSARY .....	95
INDEX .....	96



# LIST OF FIGURES

Figure 2.1 Use Case Diagram for the Existing System .....	10
Figure 2.2 Fedena School Management Software Interface .....	11
Figure 2.3 FeKara School Management Software Interface.....	12
Figure 2.4 mySkoolapp School Management Software Interface .....	13
Figure 3.1 High Level Use Case Diagram .....	24
Figure 3.2 Class Diagram for the System .....	25
Figure 3.3 Use Case Diagram for Students Registration Module.....	26
Figure 3.4 Use Case Diagram for Fees Management Module.....	26
Figure 3.5 Database Diagram for the System .....	27
Figure 3.6 User Login Interface of the System.....	29
Figure 3.7 Course Fee Payment Form .....	29
Figure 3.8 Exam Results Upload Form.....	30
Figure 4.1 Database Connection Session.....	33
Figure 4.2 Success and Error Message Function .....	33
Figure 4.3 User Login Handling Code.....	34
Figure 4.4 User Logout Message Box .....	34
Figure 4.5 Insert Operation .....	35
Figure 4.6 Update Operation.....	35
Figure 4.7 Inactivate Operation .....	36
Figure 4.8 Module Structure of the System .....	36
Figure 4.9 Re-usable Plugins .....	37
Figure 4.10 amCharts Library Files .....	38
Figure 4.11 HTML2PDF Library Files.....	38
Figure 5.1 General Comments from a User .....	45
Figure B.1 Use Case Diagram of Attendance Management Module.....	53
Figure B.2 Use case Diagram of Examination Repository Module.....	54
Figure B.3 Sequence Diagram of Students Registration Module .....	55
Figure B.4 Sequence Diagram of Course Repository Module .....	56
Figure C.1 System Login Page .....	57
Figure C.2 Password Reset Window .....	58
Figure C.3 Home Dashboard Page.....	59
Figure C.4 Navigation Menu .....	59
Figure C.5 Sub Menu Items under Main Modules .....	60

Figure C.6 Verification Message before Logout .....	60
Figure C.7 Unauthorized Access Error Display Page.....	61
Figure C.8 Success Message.....	61
Figure C.9 Failure Message .....	62
Figure C.10 Warning Message .....	62
Figure C.11 New Students Registration Form.....	63
Figure C.12 Search Students Feature.....	63
Figure C.13 View, Update, Inactivate Operations.....	64
Figure C.14 Record Attendance Procedure .....	64
Figure C.15 View Attendance Procedure .....	65
Figure C.16 Attendance Monitoring Procedure.....	66
Figure C.17 Upload Exam Results Procedure .....	67
Figure C.18 View Exam Results Procedure .....	68
Figure C.19 Course Fee Payment Procedure .....	69
Figure C.20 Course Fee Payment Invoice .....	69
Figure C.21 Counseling Fee Payment Procedure .....	70
Figure C.22 Add New Module Procedure .....	71
Figure C.23 View, Update and Inactivate Functions.....	71
Figure C.24 Payment Summary Reports .....	72
Figure C.25 Calendar Object .....	72
Figure D.1 Course Fee Payment Report Interface.....	73
Figure D.2 Course Fee Payment Monthly Report .....	73
Figure D.3 Counseling Fee Payment Report Interface .....	74
Figure D.4 Counseling Fee Payment Weekly Report.....	74
Figure D.5 Attendance per Module Report Interface .....	75
Figure D.6 Attendance per Module Monthly Report.....	75
Figure D.7 Students' Information Report Interface.....	76
Figure D.8 Student Information Monthly Report .....	76
Figure D.9 A Pie Chart Generated through System.....	77
Figure D.10 A Bar Chart Generated through System.....	77
Figure E.1 User Evaluation Questionnaire Form.....	87

# LIST OF TABLES

Table 4.1 Implementation Environment .....	31
Table 5.1 High Level Test Plan of the System .....	42
Table 5.2 Students Registration Module Test Cases .....	44
Table A.1 Hardware Requirements.....	51
Table A.2 Software Requirements .....	51
Table B.1 Use Case Description of Attendance Management Module .....	54
Table B.2 Use Case Description of Examination Repository Module .....	54
Table E.1 Test Results of User Login Module .....	78
Table E.2 Test Results of Students Registration Module .....	80
Table E.3 Test Results of Attendance Management Module .....	81
Table E.4 Test Results of Examination Repository Module.....	83
Table E.5 Test Results of Fees Management Module .....	84
Table E.6 Test Results of Course Repository Module.....	86

# LIST OF ACRONYMS

BAE	- Board of Architectural Education
B.Arch	- Bachelor of Architecture
CD-ROM	- Compact Disc Read Only Memory
CSS	- Cascading Style Sheets
ERP	- Enterprise Resource Planning
HTML	- Hypertext Markup Language
IDE	- Integrated Development Environment
OOAD	- Object-Oriented Analysis and Design
OOD	- Object Oriented Design
OOSD	- Object Oriented Software Development
PDF	- Portable Document Format
PHP	- Hypertext Pre- processor
RUP	- Rational Unified Process
SDLC	- System Development Life Cycle
SLIA	- Sri Lanka Institute of Architects
UML	- Unified Modeling Language
XAMPP	- Cross-Platform (X), Apache, MySQL, PHP, Perl

# CHAPTER 01: INTRODUCTION

## 1.1 Introduction

Architecture is both a profession and a lifestyle, relating to a broad spectrum of career opportunities. As a profession it plays a pivotal role in the production of the built environment, bridging the technical and social, practical and theoretical, the tangible and the intangible [1]. The origin of the word Architecture dates back to ancient Greece; but Architecture as a profession was already a developed art form by the time literary history began. The professional career of the Architects depends on the incredible creativity, gift to comprehend the artistic and scientific context of a given situation and the problem solving ability. Today, a professional Architect bears a social and cultural responsibility while engaged in the architectural practice.

In Sri Lanka there are only a few paths to become a Chartered Architect. As many professions, architecture has its own professional standards. The Sri Lanka Institute of Architects (SLIA) is the leading professional body for Sri Lankan architects and this institute was established in 1957 and incorporated by the Sri Lanka Institute of Architects Law No.1 of 1976. By this legislation, the SLIA was given wide powers in regard to the regulation of architectural education and development of the profession. An amendment made to the Act in 1996 gave it further powers to register chartered architects, architects and architectural licentiates who are the only persons permitted to practice the profession of architecture in Sri Lanka.

The Board of Architectural Education (BAE) of the SLIA is the education arm of the institute vested with the responsibility of organizing, supervising, and controlling the educating and training of persons to qualify as architects. Further, the BAE is required by statute to prescribe or approve courses of study for the qualifying examinations for membership of the institute, and to conduct the SLIA courses and examinations. The BAE carries on these duties manually and has found that with the increasing number of students each year, the manual practice is becoming untenable. Therefore the BAE has focused on

creating an IT based management system so that the functions of the institution can be fulfilled efficiently.

## 1.2 Motivation of the Project

Recently the BAE commenced a research programme that aims at establishing a system that could be the basis for formulation of policies and also that could assist in managing students' matters efficiently. Finally the proposed system would help to replace the present manual system progressively. Following are some of the problems faced by BAE due to not having a central system to streamline the daily processes.

- BAE is currently burdened with cumbersome paperwork and manual processes and they find it difficult to maintain proper records on students' admissions, attendance and payments.
- Lack of a proper data repository related to all examinations and courses conducted by BAE and records of assessments, marks and grades.
- BAE is currently facing difficulties in monitoring students' attendance and pending payments.
- Lack of a proper report generating tool for students' reference and management purposes.

Therefore the need of implementing a Students Management System that could help to manage students' matters efficiently and thereby reduce cost and time has become critical. Additionally, the enthusiasm for completing the final year project of the BIT degree by implementing a useful software system for SLIA was also another factor which motivated for the selection of this project.

## 1.3 Objectives of the System

The main objective of the proposed Students Management System is to provide a more organized and streamlined mechanism to manage routine processes those are currently performed manually at BAE. Following objectives are expected to be achieved through a successful implementation of the proposed system.

- Creating a centralized system for SLIA/BAE that could help to handle students' related matters in a more organized and efficient manner.
- Start handling the students' admission process through the new Students Management System after successful implementation. This includes maintaining an information repository of the students.
- Start monitoring the students' attendance through the new system. This includes sending SMS notifications through the new system when the students' attendance has not reached the required level.
- Start processing the students' payments through the new system. This functionality will include sending SMS and e-mail notifications through the new system for the payments received and payments pending. Further the students and accounts staff will be able to generate payment summary reports for the completed payments as well.
- Maintaining a data repository related to all examinations and courses conducted by the BAE. This data repository will help to improve the staff efficiency by replacing the existing manual process. This data repository will be capable of maintaining the records of examinations, assessments, marks and grades.
- Blog and a discussion forum for students, lecturers and counsellors of the institute.
- Maintaining an information repository of the Administration and Accounts staff.
- Communicating latest announcements of the SLIA through the Students Management System.
- Generating reports for students' reference and management purposes through the new system.

## 1.4 Scope of the Proposed Project

In order to successfully achieve the expected objectives, the scope of the proposed project has been defined as follows.

- Students' admission process handling system

The activities involved with students' admission process will be handled by this module. This includes maintaining an information repository of the students' as well.

- Students' payments handling and notification system

The activities involved with students' payments process can be handled by this module. Notifications will be generated for the payments received and for the pending payments as another functionality of this module. Payments summary reports for the completed payments can be generated for the students' and staff reference.

- Students' attendance handling and notification system

The activities involved with students' attendance process can be handled by this module. Sending SMS notifications to the students when the students' attendance is lagging behind the required level is another functionality of this module.

- Students' examination records handling system

The purpose of this module is to maintain a repository of data related to all the examinations conducted by the BAE. This data repository is capable of maintaining the records of examination results, assessments, marks and grades including past examination papers.

- BAE course records handling system

The purpose of this module is to maintain a repository of data related to all the courses conducted by the BAE.

- Staff information handling system

This module is for maintaining an information repository of the staff members in Administration and Accounts departments.

- Reports generating system

Following functionalities are available in this module:

- a) Generating reports for students' reference
- b) Generating reports for Management and Administration purposes

- Blog and students' discussion forum

The purpose of this module is to cater as an online forum that facilitates the discussions between students, lecturers and the counsellors of the institute.



## 1.5 Structure of the Dissertation

This dissertation provides relevant information of the Students Management System that is implemented for Sri Lanka Institute of Architects. The dissertation consists of six main chapters including the Introduction chapter and the contents of these main chapters are briefly described as follows.

### Chapter 02 – Analysis

The information related to the existing manual processes used at the client organization is described in this chapter. Requirement gathering techniques which have been used to collect system requirements as well as functional and non-functional requirements of the new system are also described in this chapter.

### Chapter 03 – Design

This chapter includes information of the system design phase such as which system design strategies have been used for system development. Further use case diagrams and database design diagrams of the proposed system as well as main interfaces of the new system are included in this chapter.

### Chapter 04 – Implementation

This chapter describes the module structure of the Students Management System and the major codes and existing codes which have been reused for the development of the modules. Hardware and software requirements, development tools and technologies used for system implementation are also explained in this chapter.

### Chapter 05 – Evaluation

This chapter consists of the information of how proper testing carried out by verifying all the aspects of the system to validate the system specification has been achieved. This includes the test plan that was used for the system validation and verification and what are the different technologies used for testing as well as user acceptance testing carried out for the system evaluation.

## Chapter 06 – Conclusion

This chapter describes a critical discussion and assessment of results of project by including the objectives of the project those have been satisfied. Lessons learnt during the project activities as well as future enhancements those can be implemented are also included in this chapter.

### References

The purpose of this chapter is to acknowledge and highlight any work of others that have been used or adapted such as all URL references and necessary citations those helped to perform project activities successfully.

### Appendices

This chapter includes additional information that is not included in the main chapters of the dissertation, however such information could be useful for the interested parties to learn more about the system.

# CHAPTER 02: ANALYSIS

## 2.1 Introduction

The Requirement Analysis phase in Software Engineering is where the project lifecycle begins. The objective of this phase is to define the system inputs, processes, outputs and interfaces in more detailed and precise manner.

IEEE defines requirements analysis as (1) the process of studying user needs to arrive at a definition of a system, hardware or software requirements. (2) The process of studying and refining system, hardware or software requirements. ‘Requirements analysis helps to understand, interpret, classify, and organize the software requirements in order to assess the feasibility, completeness, and consistency of the requirements. [2]

The details of the requirements gathering techniques those were used to collect requirements at the initial stage of developing the proposed system and how the existing system processes are functioning at BAE have been included in this chapter. Further functional and non-functional requirements of the proposed system and what are the drawbacks of the present system, are also included in this chapter.

## 2.2 Requirements Gathering Techniques

Requirements gathering process should be carefully and systematically conducted at the initial stage of designing the proposed system. The following requirements gathering techniques were used to collect the requirements of the proposed system.

- Conducting interviews

After visiting the client organization a number of interviews were conducted with the Senior Vice President of the institute in order to understand the core functions of the BAE. After that a number of interviews were conducted with the administrative staff in order to understand the business requirements and domain specific information of the organization.

- Observation

The processes those administrative staff currently follows when registering new students for the courses and collecting payments from students for various academic purposes could be understood through observing and closely monitoring real life scenarios. Additional information could be gathered by analyzing the process documents which are created with students' admission details and course payment information as well.

- Questionnaires

Examination procedures and academic modules specific information could be gathered by providing questionnaires to administrative staff and some of the office-bearers of the institute. Business requirements and other useful domain information were collected using this method and reading existing process documentation as well.

## 2.3 Analyzing the Current System and Processes

The main activities and manual processes those are currently being followed at the BAE with respect to students' academic matters can be explained as follows.

### 2.3.1 Students' Admission Process

When a new batch of students get enrolled for a new academic intake, it is required to complete a set of application forms with the students' details and submit the duly filled application forms to administrative department. Submitted enrollment applications are verified by the officers in administrative department and the admission process is concluded. Enrollment information of the students are kept in box files and stored according to the academic year.

### 2.3.2 Students' Payments Handling Process

Students' payments handling process is one of the major functions performed by Accounts department of SLIA. There are several types of payments those students are required to pay while following the courses conducted at BAE. These payment types are;

- a) Course fees
- b) Counselling fees

Students should go to the Accounts department and are required to make the relevant payments at the cashier counter and needed to obtain the student's copy of the invoice for the payment made. BAE copies of the invoices are stored in box files for future reference.

### 2.3.3 Students' Attendance Handling Process

It is compulsory to mark their attendance by signing an attendance sheet for every lecture the students participate and this is a mandatory procedure at BAE. It is a compulsory requirement to maintain at least 80% of the attendance for every module lectures and the attendance percentage will be considered when authorizing the students for the examination at the end of each academic year.

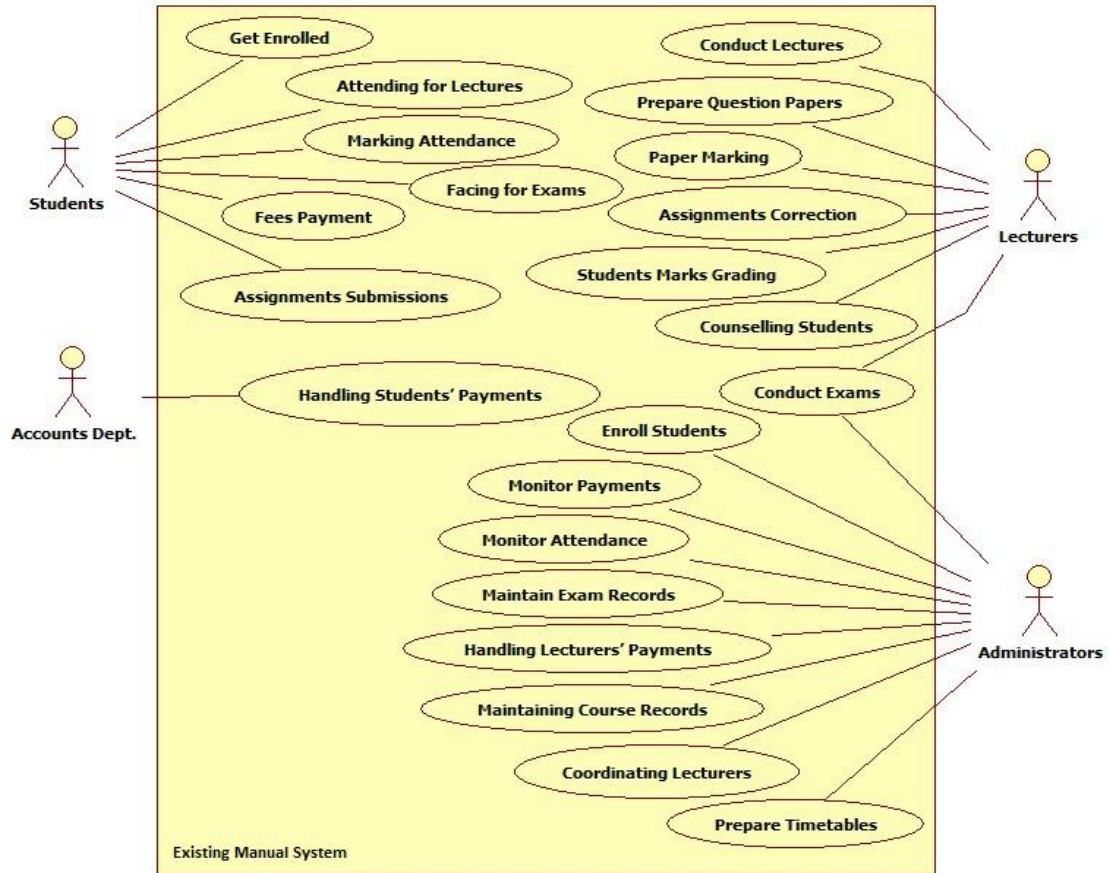
### 2.3.4 Examinations Handling Process

Examination papers are prepared by the lecturers and counsellors and exams are conducted at the end of the academic year. When the examinations are completed, answer scripts are verified by the members of the examinations board. After completing the paper marking process results are released for the students with the approval of the examinations board. Students are required to submit continuous assessments before facing year-end exams and lecturers assist in correcting the assignments submitted by students.

### 2.3.5 BAE Course Records Handling Process

The details of the course records currently conducted at BAE are maintained as PDF files by the administrative staff. The hard copies are stored in box files and can be accessible by the students, lecturers, counsellors or administrative staff of the institute.

The use case diagram of the existing manual system is depicted by Figure 2.1



**Figure 2.1 Use Case Diagram for the Existing System**

## 2.4 Existing Similar Systems Analysis

In the modern IT industry, it is not difficult task to find good and powerful web based Students Management Systems those consisting of the similar functionalities as of the proposed project. Following reviews contain comprehensive analysis of most commonly used free and open-source Students Management Systems around the world.

### 2.4.1 Fedena School Management System

Fedena is a free & open source school management software that has more features than a student information system. Fedena system can efficiently manage students, teachers, employees, courses and all the system and processes related to an education institution. [3]

Fedena School ERP System provides user-friendly dashboards with login access for teachers, non-teaching staff, students, parents and management personnel of the education institution. The various modules available in Fedena facilitate all the processes of the institution, from admission of new students to generating transfer certificates when students complete their studies. [4]

The software interface of the Fedena school management system is depicted in Figure 2.2



Figure 2.2 Fedena School Management Software Interface

#### 2.4.1.1 Pros and Cons of Fedena Software

**Pros:** Fedena offers unlimited administration and student logins to use their system, along with unlimited courses and batches. The system was developed using Ruby on Rails, so schools can easily customize the code to their school's needs. The system includes human

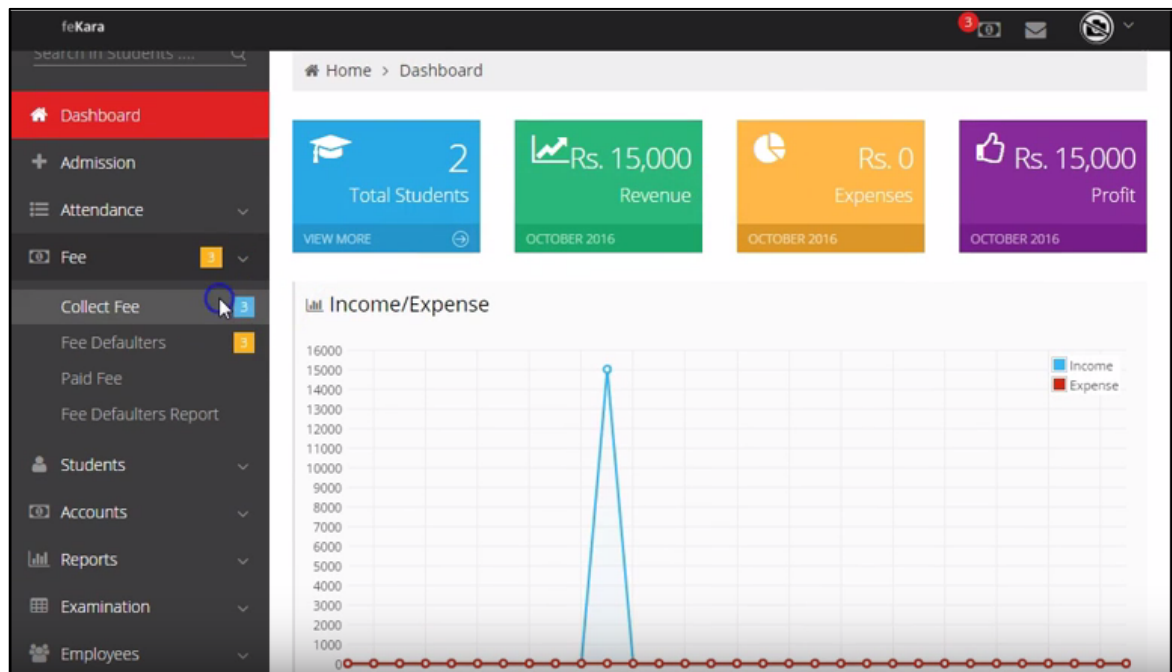
resources, a calendar, financial management, examination management, and student/parent logins. [5]

**Cons:** When comparing the free version of the Fedena software to the paid version, it becomes clear that the open-source version is lacking a number of features, including inventory, custom reports, registration and discipline. [5]

## 2.4.2 FeKara School Management System

Fekara is management software for educational institutions for learning, administration and management activities. It manages all elements of school management system including students, courses, exams and teachers, employees. Fekara has ability to manage all forms of educational institutes efficiently including, Schools/ Kindergartens, Colleges/ Universities, Tuition Centers/ Training Centers. [6]

The software interface of the Fekara school management system is depicted in Figure 2.3



**Figure 2.3 FeKara School Management Software Interface**



### 2.4.2.1 Pros and Cons of FeKara Software

**Pros:** FeKara is a clean and modern school administration and management software option which covers everything from examination and assignments to budgeting and internal messaging for all staff. FeKara even includes a mobile app which can work on all tablets and smartphones for convenient on-the-go use. [5]

**Cons:** FeKara is limited by the amount of students, bandwidth, and storage which can be managed on the free version of its software. FeKara is free for schools as long as the number of students remains under fifty students and five teachers in the system, which limits the free use of this software to much smaller schools. [5]

### 2.4.3 mySkoolapp School Management Software Mobile App

“mySkoolapp” is a state of art web based school management software system developed mainly by targeting the mobile devices that are running Android and iOS platforms. This mobile app consists with all standard functionalities that should be available in a school management system.

The software interface of the mySkoolapp system is depicted in Figure 2.4



**Figure 2.4 mySkoolapp School Management Software Interface**

### 2.4.3.1 Pros and Cons of mySkoolapp Software

**Pros:** Real time update and direct communication between teachers, principal, and parents. This software has a web based system as well. Mobile version supports iOS and Android platforms and cloud based service to access it anywhere from any device. [7]

**Cons:** Main drawback of the mobile version of this software is intermittent problems with the hosting servers are faced due to the heavy network traffic situations. [7]

## 2.5 Functional Requirements of the System

A functional requirement, in software and systems engineering, is a declaration of the intended function of a system and its components. Based on functional requirements, an engineer determines the behavior (output) that a device or software is expected to exhibit in the case of a certain input. A system design is an early form of a functional requirement. [8]

Following section describes the functional requirements that have been identified during the systems analysis and the detailed specification of the functionalities that the users expect by the proposed system.

### 2.5.1 Students' Admission Process Handling Module

It should be possible to handle functions those are associated with the new students' enrollment process using this module.

- Administrators should be able to enroll new students by providing the details of the new students.
- Administrators should be able to search and view the details of the enrolled students.
- Administrators should be able to update the details of the registered students.
- Administrators should be able to inactivate the records of the existing students when such a requirement arises.
- Generate SMS and e-mail notifications upon successful registration of the students.

### 2.5.2 Students' Payments Handling and Notification Module

The purpose of this module is handling students' payments and serving as a monitoring tool of the payments made by students. Further this module would function as a notification system that links with the students' payments. Following functions can be performed through this module.

- Accounts department should be able to accept "Course Fees" made by students and print invoice for the payments received.
- Accounts department should be able to accept "Counselling Fees" made by students and print invoice for the payments received.
- Accounts department should be able search the past payment history records made by students
- Generating SMS and e-mail alerts for the payments received from students.
- Generating SMS and e-mail reminders for the payments pending from students.

### 2.5.3 Students' Attendance Handling and Notification Module

The purpose of this module is maintaining attendance records of the students and serving as a monitoring tool for inspecting students' attendance. Further this module would function as a notification system that links with the students' attendance. Following functions are expected to be performed through this module.

- Administrators should be able to create daily attendance records of the students.
- Administrators should be able to create lecture-wise attendance records of the students.
- Administrators should have the facility to edit attendance records of the students.
- Administrators should be able to generate SMS notifications when the attendance of the students has not reached up to the target.
- History reports and graphs based on the attendance of details of the students are required to be generated on demand.

#### 2.5.4 Students' Examination Records Handling Module

The purpose of this module is maintaining a repository of data related to all the examinations conducted by the BAE. Following functions are expected to be performed through this module.

- Administrators should be able to upload examination results of the students.
- Students should be able to view the examination results.
- There should be a facility to maintain students' marks and grads.
- Students should be able to upload assignments.
- Administrators and students should be able to view uploaded assignments.
- Administrators should be able to upload past papers.
- Students should be able to view past papers.

#### 2.5.5 BAE Course Records Handling Module

The purpose of this module is to maintain a repository of data related to all the courses conducted by the BAE and following functions are expected to be performed through this module.

- Administrators should be able to create new records with course details.
- Administrators should be able to update existing course records.
- Administrators should be able to inactivate course records when such a requirement arises.
- Both students and administrators should have the facility to search and view course details.

#### 2.5.6 Staff Information Handling Module

Maintaining an information repository of the staff members in Administration and Accounts departments is the purpose of this module and following functions can be performed through this module.

- Administrators should have facility to create new records by providing staff information.

- Administrators should be able to search and view the details of staff members.
- Administrators should be able to update the details of the existing staff members.
- Administrators should be able to inactivate the records of the existing staff members when such a requirement arises.

### 2.5.7 Reports Generating Module

The purpose of this module is generating reports that facilitate following requirements.

- Generating reports for students' reference
- Generating reports for Management and Administration purposes

### 2.5.8 Blog and Students Discussion Forum

The purpose of this module is to accommodate as an online forum that facilitates the discussions between students, lecturers and counsellors of the institute.

- Students should be able to update the forum with the questions they have.
- Lecturers or counsellors should be able to update the forum with the answers for the questions those the students have posted.
- Students also should be able to update the forum with the answers for the questions those the other students have posted.

## 2.6 Non-functional Requirements of the System

The definition for a non-functional requirement is that it essentially specifies how the system should behave and that it is a constraint upon the systems behavior. One could also think of non-functional requirements as quality attributes of a system. Non-functional requirements cover all the remaining requirements which are not covered by the functional requirements. They specify criteria that judge the operation of a system, rather than specific behaviors. [9]

Following section explains about the non-functional requirements that should be focused when implementing the proposed Students Management System.

### 2.6.1 Accuracy

Since the proposed system intends to deal with marks and grades those the students have obtained for the examinations, those information should be accurate and trustworthy. Also the notifications that would be generated from the payments handling and attendance handling modules should be accurate and up-to-date.

### 2.6.2 Security

Since the system contains sensitive information such as fees payments made by students and marks and grades those the students have obtained for examinations, securing such information will be one of the major non-functional requirements that should be focused. Only the authorized people should be able to access the system using user names and passwords. A mechanism for encrypting the passwords should also be implemented in order to secure the user accounts that registered in the system. Proper access privileges should be defined for the different user roles.

### 2.6.3 Reliability

Availability of the system and system uptime should be maintained up to the maximum level that is agreed with the client.

### 2.6.4 User-friendliness

Overall functionalities of the system should be implemented in a way that is easy to learn and easy to use for all the user levels who take benefits of the system.

### 2.6.5 Maintainability

Database backups should be scheduled and taken at defined intervals in order to protect the system from unexpected data loss and system failure situations.

## 2.7 Overview of Software Process Models

The decision of selecting the most suitable software process model for developing the intended software is usually considered as one of the most challenging decisions any software engineer has to make. That is mainly because the success factor of the developed

software mostly depends on the decision of selecting the most suitable software process model. Therefore it's required to perform a comprehensive analysis about the 'Requirements of the project' before taking the final decision on choosing among the structured methodologies and object oriented methodologies for software development.

If the software project consists of a set of very clear user requirements and if requirements are not changing throughout the development life cycle, selecting a structured process models such as 'Waterfall model' and 'Spiral model' would be most the suitable decision for the success of the software project.

On the other hand for software projects with unclear user requirements and if requirements are changing throughout the development life cycle, selecting an object oriented methodology such as "Agile model" or "Iterative and Incremental model" would be the most suitable decision for the success of the project.

However focusing on the requirements of the project is only one technique for selecting the suitable process model for software development. There are several other important factors that should be evaluated such as time frame of the project, budget limitation, skills limitation and component reusability when considering the process model that is most suitable for the software project.

## 2.8 Selected Methodology for the Proposed System

After considering several software development methodologies, the Rational Unified Process (RUP) was selected as the most suitable process model for developing the proposed Students Management System.

Rational Unified Process is a software development process from Rational, a division of IBM. It divides the development process into four distinct phases that each involves business modeling, analysis and design, implementation, testing, and deployment. [10]

The main reason for this selection is, RUP has the supportability for Object Oriented Software Development (OOSD). Further RUP is considered as an iterative and incremental software methodology and this model has the acceptance of use cases and Unified Modeling Language (UML). As the proposed system consists of several modules

such as Admission module and Fees management module, etc. each module can be implemented iteratively and incrementally by dealing with the changing requirements throughout the development life cycle.

RUP methodology consists of four phases and following section contains a brief description of each phase. [10]

- **Inception Phase**

The idea for the project is stated. The development team determines if the project is worth pursuing and what resources will be needed.

- **Elaboration Phase**

The project's architecture and required resources are further evaluated. Developers consider possible applications of the software and costs associated with the development.

- **Construction Phase**

The project is developed and completed. The software is designed, written, and tested.

- **Transition Phase**

The software is released to the public. Final adjustments or updates are made based on feedback from end users.



# CHAPTER 03: DESIGN OF SOLUTION

## 3.1 Introduction

Systems design is the process of defining elements of a system like modules, architecture, components and their interfaces and data for a system based on the specified requirements. It is the process of defining, developing and designing systems which satisfies the specific needs and requirements of a business or organization. [11]

## 3.2 Alternative Solutions for the Proposed System

When it comes to the evaluation of different competing design strategies those are applicable for the proposed system, following alternative strategies can be mainly emphasized as relevant.

### 3.2.1 Purchasing a Commercial Product from a Software Vendor

One of the main alternative solutions would be purchasing a commercially available Student Management Software for BAE and customize the new system according to the requirements of the institute. Even though this strategy seems to be a straight forward solution, following disadvantages have been identified in this strategy.

- a) Commercially available software is usually priced very high and the client organization is reluctant to go with this solution due to this matter.
- b) Even though a generic Student Management System can be purchased from a vendor, the customization process according to the specific requirements is not straightforward and this process usually takes a long time.
- c) Obtaining after sales software support for the commercial products is a cost involved process.

Even though the requirements of the proposed project could be satisfied through this strategy, client organization is not willing to go with this solution and it was decided to exclude this method due to the above drawbacks involved in this strategy.

### 3.2.2 Implementing the System as a Stand-alone Application

The next alternative strategy that can be used to design the proposed system is implementing the solution as a stand-alone desktop application. Even though a majority of the objectives of the proposed project could be satisfied using this strategy, it was decided to exclude this method from implementation due to the following drawbacks involved with this option.

- a) One of the major drawbacks of the standalone application would be platform dependency since this method requires a local runtime environment such as Windows.
- b) The cost of hardware and software that is required for the implementation of this strategy is comparatively high.
- c) Access for the system can only be obtained while the users are in the office premises.
- d) Installation of the desktop application is required to be done on several computers. Updating and maintenance of the application is also a difficult procedure since it is required to be done on every single computer that the application has been installed.
- e) Since the system is to be used by the BAE and the finance department that is also handling the main financial activities of the SLIA, the stand-alone system would not be suitable.
- f) This method of implementation is not the right selection for the software projects with short time duration.

### 3.3 Implementing the System as a Web Based Application

After conducting a comprehensive analysis on the alternative design strategies that the proposed system could be implemented, it was decided unanimously to design and implement the system as a web based application. There are many positive arguments that supported this decision and following are the most important motives those can be highlighted among them.

- a) Flexible core technologies and open source platforms such as PHP and MySQL can be used for developing the web based applications.
- b) Cost effective development and troubleshooting process is much easier than in stand-alone desktop application.
- c) System installation and maintenance is less complicated in web based applications.
- d) Web based applications can be easily customizable by reusing the development frameworks such as bootstrap templates.
- e) Web based applications can be easily accessible from anywhere without reaching to the office premises.
- f) There is a possibility for future enhancements by configuring the accessibility for a wide range of mobile devices such as smart phones and tablets.

### 3.4 Object-Oriented Systems Analysis and Design

Object-Oriented Analysis and Design (OOAD) is a technical approach used in the analysis and design of an application or system through the application of the object-oriented paradigm and concepts including visual modeling. This is applied throughout the development life cycle of the application or system, fostering better product quality and even encouraging stakeholder participation and communication. [12]

UML is the standardized notation and modeling language that support systems developers to visualize, construct and document the artifacts of a software system during system design phase. There are two main categories of UML diagrams according to the standard UML notation.

- 1) Structure diagrams
- 2) Behavioral diagrams

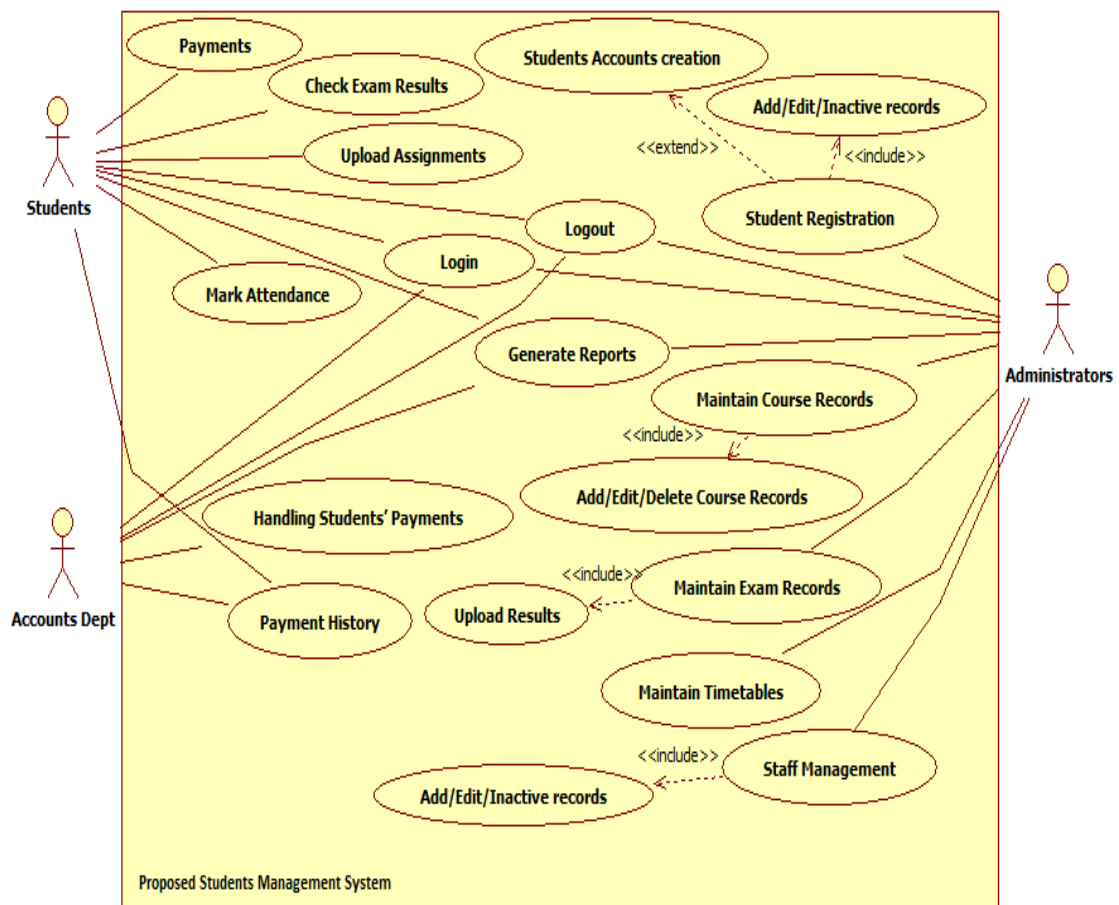
The main types of Structure diagrams those are most commonly used are Class Diagram, Component Diagram and Object Diagram whereas the main types of Behavioral diagrams those are most commonly used are Use Case Diagram, Activity Diagram and Sequence Diagram in software design.

The following section describes the structure and behavior of the Students Management system according to the standard UML notation

### 3.4.1 High Level Use Case Diagram for the Students Management System

Use case diagrams provide a graphic overview of the actors involved in a system and different functions performed by those actors and how these different functions interact with the system.

The high level use case diagram for the proposed Students Management System is depicted in Figure 3.1



**Figure 3.1 High Level Use Case Diagram**

### 3.4.2 Class Diagram for the Students Management System

The Class Diagram shows the classes in a system, attributes, and operations of each class and the relationship between each class.

Class Diagram for the proposed Students Management System is depicted in Figure 3.2

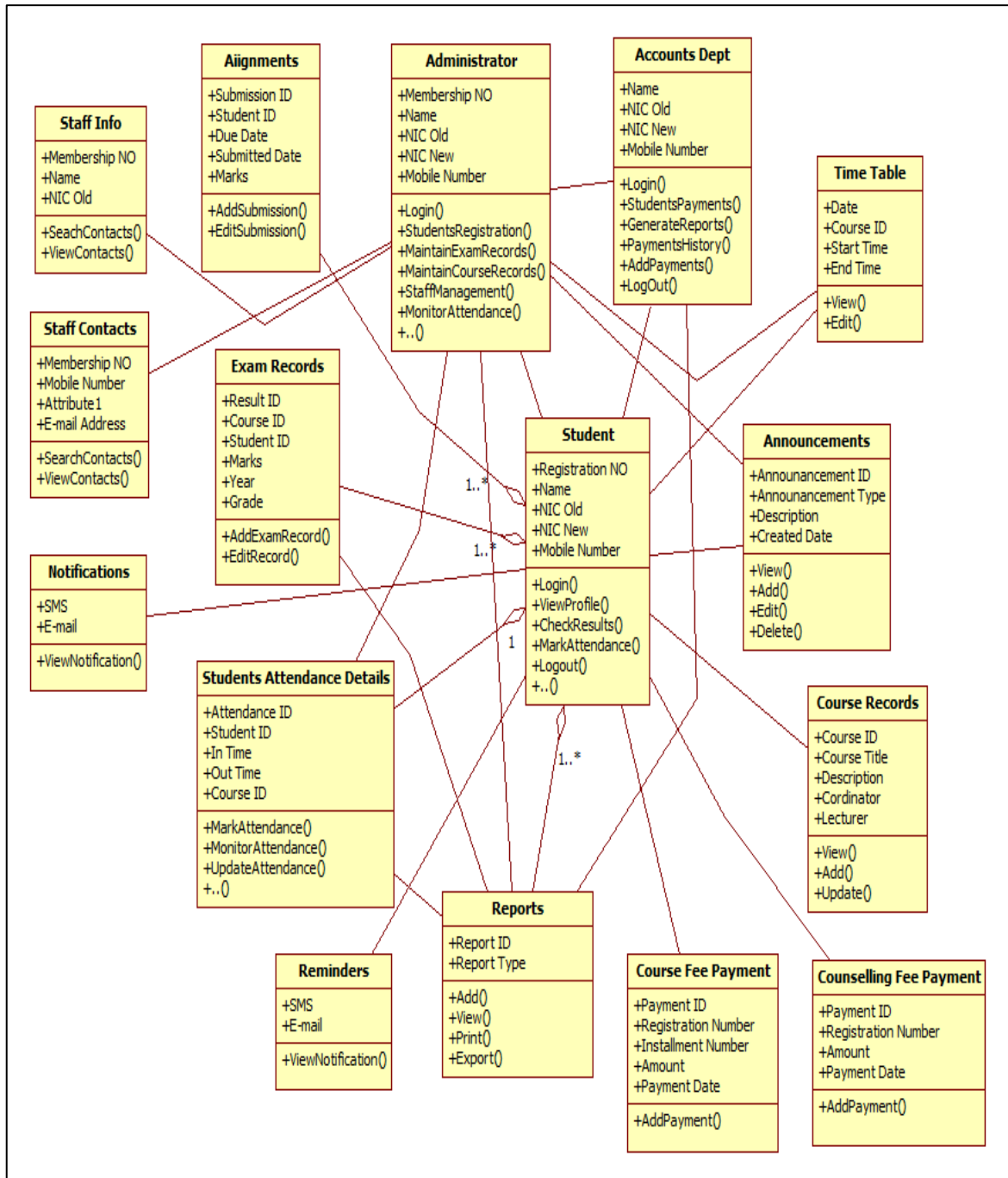
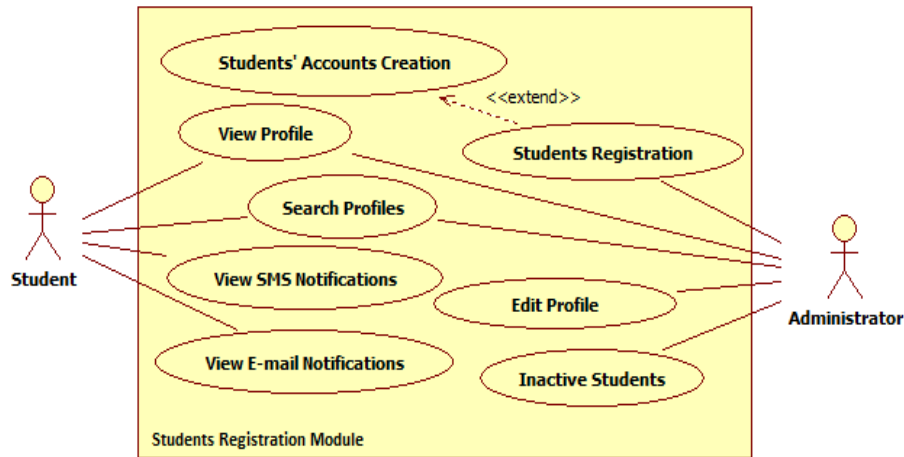


Figure 3.2 Class Diagram for the System

### 3.4.3 Use Case Diagram for the Students Registration Module

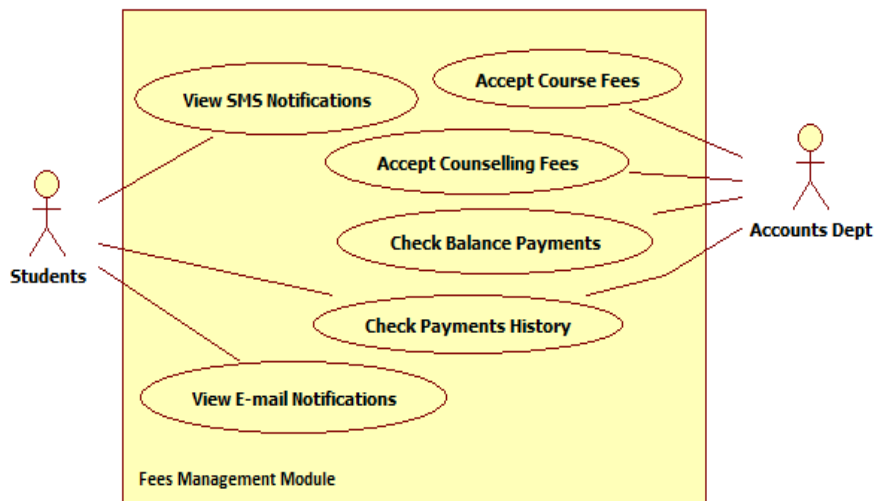
The actors involved in students registration module and the functions performed by those actors are depicted in Figure 3.3



**Figure 3.3 Use Case Diagram for Students Registration Module**

### 3.4.4 Use Case Diagram for the Fees Management Module

The actors involved in fees management module and the functions performed by those actors are depicted in Figure 3.4



**Figure 3.4 Use Case Diagram for Fees Management Module**

### 3.5 Database Design for the System

Designing a good database model that conveniently helps to achieve the functional requirements of the system is one of the major tasks in the Design phase. There are two main database design objectives those are expected to be achieved in a good database model.

- 1) Data redundancy elimination
- 2) Ensuring data integrity and accuracy

Normalization is the database design approach that can be applied to achieve above mentioned objectives. Normalization process comprises several stages and every stage has different goals to be achieved. When designing the database model for the proposed system, First Normal Form (1NF), Second Normal Form (2NF) and Third Normal Form (3NF) should be considered to create a database that has achieved above mentioned design objectives.

Database diagram created for the Students Management database is depicted in Figure 3.5

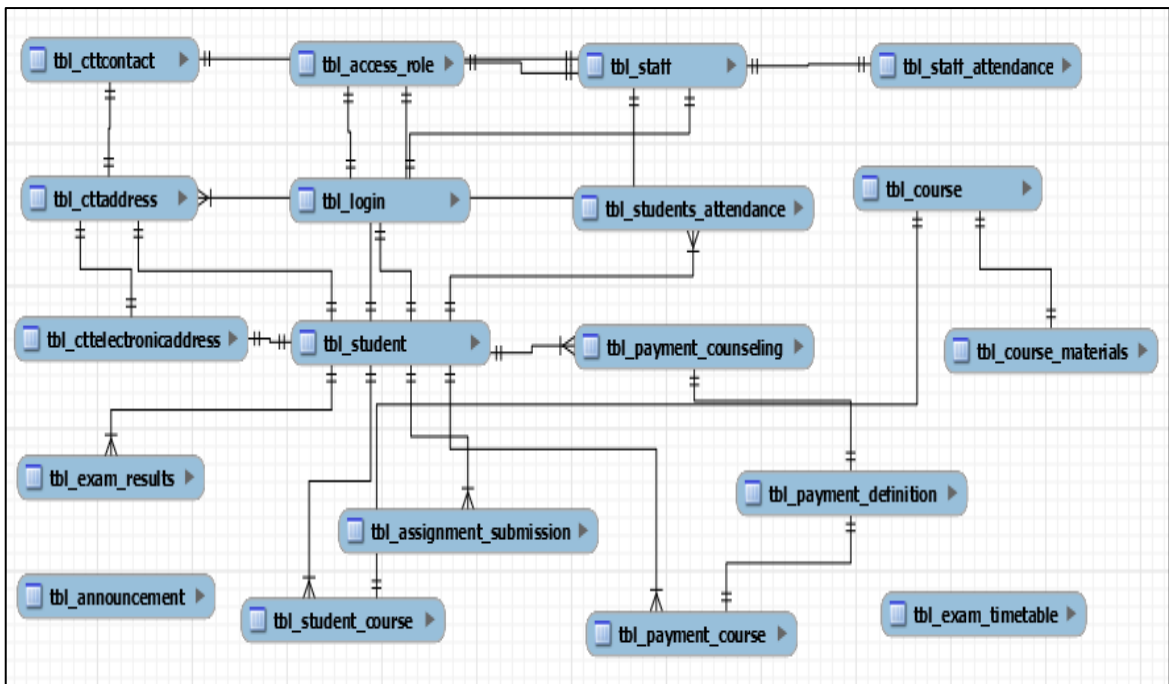


Figure 3.5 Database Diagram for the System

### 3.6 User Interface Design for the System

User interface design also known as the front-end design of a software system is one of the crucial phases that should be sensibly carried-out in the software design stage. The intended functionalities of the software system are carried out by the user with the help of the controls available in the user interfaces. In that sense user interfaces facilitate as the fundamental platform for the interaction between human and computer system.

There are many fundamental principles that should be considered when designing user interfaces for a software solution and following are the most important points among them.


- The structure  
The design of the user interface should be meaningful and purposeful and it should be applicable to the software system.
- Simplicity  
The design of the user interface should be as simple as possible and not complicated for the users.
- Consistency  
User should feel familiar with the system after starting to work with the system.
- Efficiency  
The design of the user interface should facilitate the tasks of the software system in the best possible way.
- Tolerance  
The design of the user interface should be flexible and tolerant of the mistakes done by users. Also the system users should not be surprised by seeing the behavior of the system.

These fundamental user interface design principles have been properly considered when designing the user interfaces of the Students Management System. All the user interfaces of the system have been created in a simple, consistent and creative manner.



### 3.6.1 User Login Interface

Figure 3.6 shows the user login page that initially loads when the users start the system. This is a common user interface for all the users such as Students, Administrators and Accounts department and only the authorized users have access to the system.

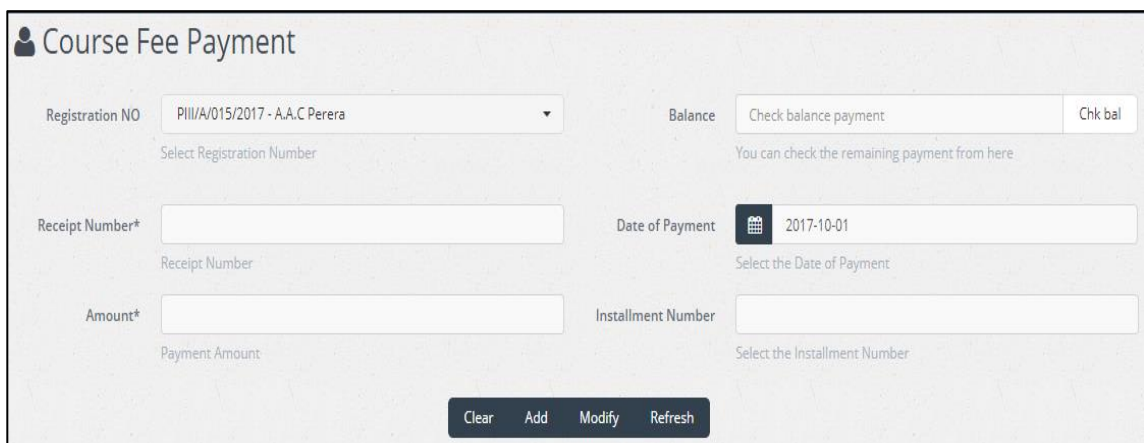


The image shows a user login interface. At the top, there is a red banner with the text "SLIA" in white. Below this, a dark grey box contains the text "Welcome to BAE, Please login". Underneath, there are two input fields: "Username" and "Password". To the left of the "Password" field is a link that says "Forgot your password?". To the right of the "Password" field is a blue button labeled "Log In". Below the login form, there is a white banner with the text "Sri Lanka Institute of Architects" and a circular logo on the right side.

**Figure 3.6 User Login Interface of the System**

### 3.6.2 Course Fee Payment Interface

The user interface of the course fee payment form is shown in Figure 3.7



The image shows a course fee payment form. At the top left, there is a user icon and the text "Course Fee Payment". Below this, there are several input fields and buttons. On the left, there is a "Registration NO" dropdown menu with the value "PIII/A/015/2017 - A.A.C Perera" and a "Select Registration Number" label. Below it is a "Receipt Number\*" input field with a "Receipt Number" label. Below that is an "Amount\*" input field with a "Payment Amount" label. On the right, there is a "Balance" section with a "Check balance payment" button and a "Chk bal" button. Below this is a "Date of Payment" input field with a calendar icon and the value "2017-10-01" and a "Select the Date of Payment" label. Below that is an "Installment Number" input field with a "Select the Installment Number" label. At the bottom, there is a dark grey button with the text "Clear Add Modify Refresh".

**Figure 3.7 Course Fee Payment Form**

### 3.6.3 Exam Results Upload Interface

The user interface of the exam results upload form is shown in Figure 3.8

Year	Registration Number	Module Code	Marks	Comment	Effective Date	Approved By
2017	PIII/A/015/2017	AB123	80		2017-10-01	BAE Approved ▼
2017	PIII/A/015/2017	AB12312	60		2017-10-01	BAE Approved ▼
2017	PIII/A/015/2017	AB1231ASD	45		2017-10-01	BAE Approved ▼
2017	PIII/A/015/2017	AB123C	77		2017-10-01	BAE Approved ▼
2017	PIII/A/015/2017	AB123B	86		2017-10-01	BAE Approved ▼
2017	PIII/A/015/2017	XY123	79		2017-10-01	BAE Approved ▼

**Figure 3.8 Exam Results Upload Form**

# CHAPTER 04: IMPLEMENTATION

## 4.1 Introduction

Implementation phase of the SDLC is the activity where user requirements are developed using the selected programming languages and tools according to the requirements specification. This is the activity where functionalities expected by system users become executable functions. This phase is executed gradually and carefully from one stage to the next by implementing the functions that have been defined as several modules.

This chapter describes the major code segments that have been used and the modular structure of the system. Further this chapter includes details of the hardware and software implementation environment along with proper acknowledgments of the existing codes reused for the system implementation.

## 4.2 Implementation Environment

The Hardware and Software configurations that were used for system implementation have been categorized in the Table 4.1

Hardware Specification	Software Specification
IntelI CoreI i5-4200M CPU @ 2.50GHz	Microsoft Windows 7 Professional
8GB RAM	XAMPP Control Panel v3.2.1
500GB Hard Disk	Apache Server
	MySQL 5.6.24
	PHP Version 5.6.8

**Table 4.1 Implementation Environment**

## 4.3 System Development Tools and Technologies

### 4.3.1 NetBeans IDE 8.2

NetBeans IDE 8.2 was used as the IDE for project implementation. All front-end implementation tasks that involved PHP, HTML5 and CSS coding were carried out using this user friendly development tool in a convenient manner.

### 4.3.2 MySQL Workbench 6.3

The open source relational database management system, MySQL was used for database management tasks throughout the implementation of the project and MySQL Workbench 6.3 tool was used for performing visual database designing and modeling activities.

### 4.3.3 XAMPP Server v3.2.1

XAMPP, (stands for Cross platform, Apache HTTP Server, MySQL, PHP and Perl) which is an open source cross-platform web server solution. XAMPP is a simple and lightweight web server solution which is very useful for implementing and testing web based systems.

### 4.3.4 StarUML 5.0.2

The software design diagrams which included in the dissertation were created using StarUML 5.0.2 open-source UML design tool.

## 4.4 Major Code Structure

This section describes all major code segments that have been implemented during the Implementation phase to make the intended functionalities of the project a workable and productive solution.

### 4.4.1 Database Connection Session

Defining the database connection that bridge the connectivity with the application is very important for proper functioning of the implemented system. The code segment depicted in Figure 4.1 describes the MySQL connection string that has been defined to connect with the “db\_bae” database.

```

<?php

session_start(); //Initiate a database session

/** DATABASE CONFIGURATION */
define('DB_HOST', 'localhost');
define('DB_PORT', '3306');
define('DB_NAME', 'db_bae');
define('DB_USER', 'root');
define('DB_PASS', '');

/** DATABASE CONNECTION-STRING */
$conn = new PDO('mysql:host=' . DB_HOST . ';port=' . DB_PORT . ';dbname=' . DB_NAME, DB_USER, DB_PASS);
$conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION); // This code handles errors

/** SITE CONFIGURATION */
define('BASE_URL', 'http://localhost/BAE/');
define('LOCAL_PATH', 'D:\xampp\htdocs\BAE\');
define('SITE_TITLE', 'Student Management System');

require_once 'template.php';
?>

```

**Figure 4.1 Database Connection Session**

#### 4.4.2 Success and Error Messages

The code segment depicted in Figure 4.2 describes how “Success” and “Error” messages have been implemented in the code base.

```

public function showMsg() { // This function will trigger success and error messages
    if (isset($_SESSION['SUCESSS'])) { // This will trigger SUCESSS messages
        foreach ((array) $_SESSION['SUCESSS'] as $SUCESSS) {
            echo "<div class='alert alert-success alert-dismissible fade in' role='alert'>
                <strong> $SUCESSS</strong>
            </div>";
            unset($_SESSION['SUCESSS']);
        }
    }

    if (isset($_SESSION['ERROR'])) { // // This will trigger ERROR messages and user warning messages
        foreach ((array) $_SESSION['ERROR'] as $ERROR) {
            echo "<div class='alert alert-danger alert-dismissible fade in' role='alert'>
                <strong>$ERROR </strong>
            </div>";
            unset($_SESSION['ERROR']);
        }
    }
}

```

**Figure 4.2 Success and Error Message Function**

#### 4.4.3 User Login Handling Code

The code segment depicted in Figure 4.3 handles the user login accounts in the system. An encryption function called “sha1” has been used to secure the passwords of the registered user accounts by applying this encryption algorithm.

```

try {
// Check the user account from database
$sql = 'SELECT User_Name,Password,User_Role FROM db_bae.tbl_login where User_Name=:User_Name ';
$stmt = $conn->prepare($sql);
$stmt->execute(array(':User_Name' => $User_Name));
$result = $stmt->fetchAll();

if (count($result)) {
    $row = $result[0];
    $DBPassword = $row['Password']; // Verify database password
    $_SESSION['username'] = $row['User_Name'];
    $_SESSION['User_Role'] = $row['User_Role'];

    if ($DBPassword == sha1($Password)) { // verify user input password with the database password
        if ($row['User_Role'] == 'Administrator') { //Administrator User Role
            $sql2 = 'SELECT Name_With_Initials,NIC_Old,Staff_Image FROM db_bae.tbl_staff where NIC_Old=:NIC_Old';
            $stmt2 = $conn->prepare($sql2);
            $stmt2->execute(array(':NIC_Old' => $User_Name));
            $result = $stmt2->fetchAll();
            $row = $result[0];
            $_SESSION['profile_Image'] = $row['Staff_Image'];
            $_SESSION['Name_With_Initials'] = $row['Name_With_Initials'];
        } if ($row['User_Role'] == 'Student') { // Student User Role
            $sql3 = 'SELECT Name_With_Initials,NIC_Old,Student_Image FROM db_bae.tbl_student where NIC_Old=:NIC_Old';
            $stmt3 = $conn->prepare($sql3);
            $stmt3->execute(array(':NIC_Old' => $User_Name));
            $result = $stmt3->fetchAll();
            $row = $result[0];
            $_SESSION['profile_Image'] = $row['Student_Image'];
            $_SESSION['Name_With_Initials'] = $row['Name_With_Initials'];
        } else {
            $sql4 = 'SELECT Name_With_Initials,NIC_Old,Staff_Image FROM db_bae.tbl_staff where NIC_Old=:NIC_Old';
            $stmt4 = $conn->prepare($sql4);
            $stmt4->execute(array(':NIC_Old' => $User_Name));
            $result = $stmt4->fetchAll();
        }
    }
}

```

Figure 4.3 User Login Handling Code

#### 4.4.4 Logout Message Box

The code segment depicted in Figure 4.4 handles the user logout function and the message box that triggers for the logout event.

```

public function getLogoutMag() { //Logout Message function
    echo '
<!-- MESSAGE BOX-->
<div class="message-box animated fadeIn" data-sound="alert" id="mb-signout">
    <div class="mb-container">
        <div class="mb-middle">
            <div class="mb-title"><span class="fa fa-sign-out"></span> Log <strong>Out</strong> ?</div>
            <div class="mb-content">
                <p>Are you sure you want to log out?</p>
                <p>Press No if youwant to continue work. Press Yes to logout current user.</p>
            </div>
            <div class="mb-footer">
                <div class="pull-right">
                    <a href="'. BASE_URL . 'apps/model/Login/_logout.php" class="btn btn-success btn-lg">Yes</a>
                    <button class="btn btn-default btn-lg mb-control-close">No</button>
                </div>
            </div>
        </div>
    </div>
</div>
<!-- END MESSAGE BOX-->
';
}

```

Figure 4.4 User Logout Message Box

#### 4.4.5 Insert, Update and Inactivate Code Segments

Insert records to the database, update existing records and inactivate some records in the database are some of the major functions in the system. Figure 4.5 illustrates insert operation and Figure 4.6 illustrates update operation functions are implemented in codebase.

- Insert Operation

```
//-----Validate the Amount to be paid

if ($Paid > $TotAmt) {
    $_SESSION['ERROR'][] = 'Error !!, Amount to be paid ' . $TotAmt.' Total Amount to pay ' . $Paid;
} else { // Insert the Course Payment Amount to table
    $sql = 'INSERT INTO tbl_payment_course(Registration_Number,Installment_Number,Receipt_Number,Amount,Payment_Date)
        . 'VALUES(:Registration_Number,:Installment_Number,:Receipt_Number,:Amount,:Payment_Date)';

    $stmt = $conn->prepare($sql);
    $stmt->execute(array('Registration_Number' => $Registration_Number,
        'Installment_Number' => $Installment_Number,
        'Receipt_Number' => $Receipt_Number,
        'Amount' => $Amount,
        'Payment_Date' => $Payment_Date));
    $conn = null;

    $_SESSION['SUCCESS'][] = 'New Course Payment Added Successfully: Registration Number ' . $Registration_Number. ' .
```

**Figure 4.5 Insert Operation**

- Update Operation

```
try {
    if (isset($_POST['Modify'])) { // This modifies the details of course modules

        $sql = 'UPDATE tbl_course '
            . 'SET Module_Code=:Module_Code, Course_Title=:Course_Title, Testing_Mode=:Testing_Mode,
WHERE Module_Code=:Module_Code';

        $stmt = $conn->prepare($sql);
        $stmt->execute(array('Module_Code' => $Module_Code,
            'Course_Title' => $Course_Title,
            'Testing_Mode' => $Testing_Mode,
            'Course_Description' => $Course_Description,
            'Course_Coordinator' => $Course_Coordinator,
            'Lecturer' => $Lecturer,
            'Examiner' => $Examiner,
            'NumberOfSessions' => $NumberOfSessions,
            'MinimumSessions' => $MinimumSessions));
        $conn = null;

        $_SESSION['SUCCESS'][] = 'Module Details Successfully Updated : ' . $Module_Code;
    }
}
```

**Figure 4.6 Update Operation**

- Inactivate Records Operation

Figure 4.7 illustrates the code segment of inactivating database records.

```
try { // This inactivates the Course Module record in database

    $sql = "UPDATE tbl_course SET Is_Active='Inactive' WHERE Module_Code=:Module_Code";

    $stmt = $conn->prepare($sql);
    $stmt->execute(array('Module_Code' => $Module_Code));

    $_SESSION['SUCESSS'][] = 'Module Details Successfully Inactivated : ' . $Module_Code;

} catch (Exception $ex) {
    $_SESSION['ERROR'][] = $ex->getMessage();
}
```

Figure 4.7 Inactivate Operation

## 4.5 Module Structure of the System

Figure 4.8 illustrates the module structure of the system

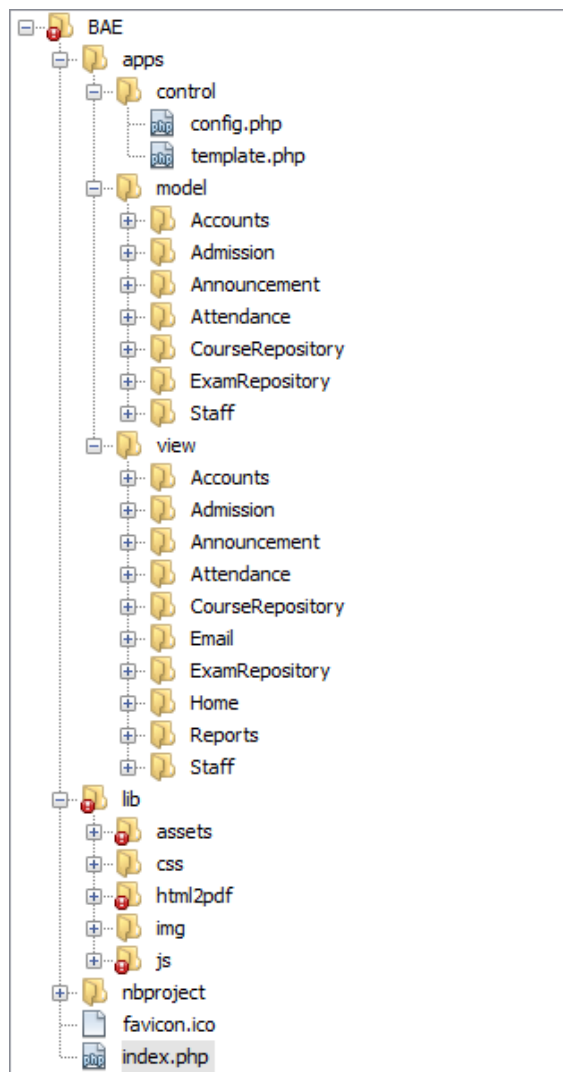


Figure 4.8 Module Structure of the System



## 4.6 Re-Usable Existing Components

Following section describes the re-usable components which have been used throughout the implementation of the system.

- **Bootstrap Admin Template**

“Atlant – Responsive Bootstrap Admin Template by Aqvatarius” has been used as the framework of developing the system. Even though this Bootstrap template was used as the system framework, all the implementation work of the system modules were designed and developed from the scratch according to the requirements of the client.

- **Library Files**

The default library files such as JavaScript and jQuery plugins that are available with the “Atlant - Bootstrap Admin Template” have been used for the system development in order to be compatible with the base framework that was chosen for implementation. Figure 4.9 illustrates the JavaScript and jQuery plugins that were used as the re-usable components during implementation.

```
<!-- START PLUGINS -->
<script type="text/javascript" src = "" . BASE_URL . 'lib/js/plugins/jquery/jquery.min.js"></script>
<script type="text/javascript" src = "" . BASE_URL . 'lib/js/plugins/jquery/jquery-ui.min.js"></script>
<script type="text/javascript" src = "" . BASE_URL . 'lib/js/plugins/bootstrap/bootstrap.min.js"></script>
<!-- END PLUGINS -->

<!-- THIS PAGE PLUGINS -->
<script type="text/javascript" src = "" . BASE_URL . 'lib/js/plugins/ichack/ichack.min.js"></script>
<script type="text/javascript" src = "" . BASE_URL . 'lib/js/plugins/mcustomscrollbar/jquery.mCustomScrollbar.min.js"></script>

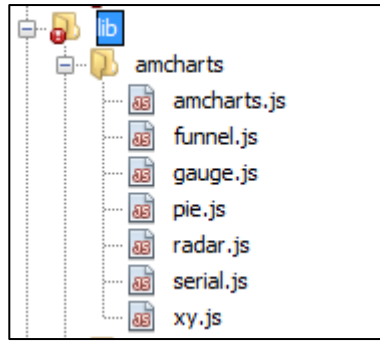
<script type="text/javascript" src = "" . BASE_URL . 'lib/js/plugins/bootstrap/bootstrap-datepicker.js"></script>
<script type="text/javascript" src = "" . BASE_URL . 'lib/js/plugins/bootstrap/bootstrap-select.js"></script>

<!-- END THIS PAGE PLUGINS -->
```

**Figure 4.9 Re-usable Plugins**

- **amCharts Library Files**

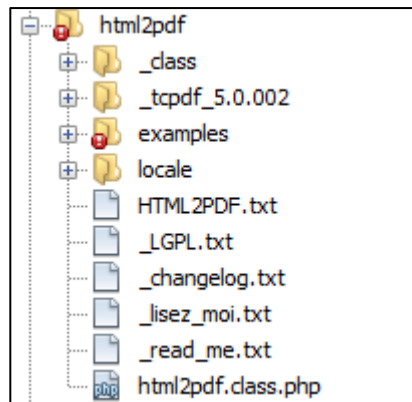
The reusable library component called “amCharts” was used for data visualization and creating charts and graphs through the system. Figure 4.10 illustrates of amCharts reusable library files.



**Figure 4.10 amCharts Library Files**

- **HTML2PDF Library Files**

HTML2PDF reusable library files written in PHP were used for handling conversion mechanism of HTML files to PDF format in the system. Figure 4.11 illustrates HTML2PDF library files which were used in the system.



**Figure 4.11 HTML2PDF Library Files**

The original author of the “Atlant – Responsive Bootstrap Admin Template” is Aqvataris [13] and the team Aqvataris should be thankfully acknowledged. However this Bootstrap Admin Template was used only as the framework of the Student Management System and all the modular wise development work of the system was carried out from the beginning according to the client’s requirements.

# CHAPTER 05: EVALUATION

## 5.1 Introduction

Software testing is primarily a broad process that is composed of several interlinked processes. The primary objective of software testing is to measure software health along with its completeness in terms of core requirements. Software testing involves examining and checking software through different testing processes. The objectives of these processes can include:

- Verifying software completeness in regards to functional/business requirements.
- Identifying technical bugs/errors and ensuring the software is error-free.
- Assessing usability, performance, security, localization, compatibility and installation. [14]

## 5.2 Software Testing vs Software Evaluation

Software Testing and Software Evaluation are two major processes that should be clearly understood in order to produce a high quality software system.

### 5.2.1 Software Testing Process

The objective of the software testing process is to verify that the software system performs according to its design specification. There are different techniques that can be used for software testing and following are the most popular and widely used methods among them.

#### 1) Black-Box Testing

The technique of testing without having any knowledge of the interior workings of the application is called black-box testing. The tester is oblivious to the system architecture and does not have access to the source code. [15]

Black-Box testing technique was performed successfully with the help of few students who are unaware to the system architecture and the user inputs were documented properly.

## 2) White-Box Testing

White-box testing is the detailed investigation of internal logic and structure of the code. In order to perform white-box testing on an application, a tester needs to know the internal workings of the code. [15]

### 5.2.1.1 Types of Testing

There are many types of software testing also known as ‘Levels of testing’ that are used to validate the usability of a software system.

#### 1) Unit Testing

A level of the software testing process where individual units or components of a software system are tested. The purpose is to validate that each unit of the software performs as designed. [16]

#### 2) Integration Testing

A level of the software testing process where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. [16]

#### 3) System Testing

A level of the software testing process where a complete, integrated software system is tested. The purpose of this test is to evaluate the system’s compliance with the specified requirements. [16]

#### 4) Acceptance Testing

A level of the software testing process where a system is tested for acceptability. The purpose of this test is to evaluate the system’s compliance with the business requirements and assess whether it is acceptable for delivery. [16]

Unit testing was performed in parallel with each individual module development during system implementation phase. Integration and System testing activities were carried out after completing the system implementation with the purpose of evaluating the system’s compliance with the user requirements those were collected at the Analysis phase.

### 5.2.2 Software Evaluation

The objective of software evaluation process is to validate that the implemented software system fulfills the requirements of the project. One method of software evaluation is to use the potential end users to evaluate the usability of the system.

### 5.3 System Test Plan

A Test Plan can be described as a methodical and systematic approach which is used to verify and validate the quality of a software system. Defining a good test plan and a strategy can be very effective in discovering errors and bugs of the system. A properly defined and completed test plan can be used as documentary evidence to prove the quality of the software product. The Table 5.1 shows the high level test plan that is defined to validate and verify the Students Management System.

Module Name	Functions To Be Tested	Test Priority
User login module	Student login function	High
	Staff login function	High
	Forgot your password function	High
	Disable user account after 5 unsuccessful attempts	Medium
Students registration module	New students registration function	High
	Search registered students	Medium
	View students' profile details	Medium
	Update students' profile details	High
	Inactive records of the existing students	High
Attendance management module	Record attendance of the students	High
	View attendance of the students	High
	Monitor attendance of the students	High
Examination repository module	Upload exam results of the students	High
	View exam results of the students	High
	Assignments submission	High

	View submitted assignments	High
	Upload past papers	High
	View uploaded past papers	High
Course repository module	Add the details of a new course module	High
	View the details of the added modules	Medium
	Update the details of the existing modules	High
	Inactive course records in Course Repository	High
Fees management module	Add course fee payment	High
	Add counseling fee payment	High
	Search payment history records	High
	Monitor due payments function	High
Staff management module	Add new staff member function	High
	Search existing staff members	Medium
	View staff member profile details	Medium
	Update staff member profile details	High
	Inactive records of the existing staff members	High
Reporting Module	Payment information Reports	High
	Attendance details reports	High
	Students' information reports	High
Announcement Module	Publish new announcements	High
	Edit announcements	Medium
	Inactivate announcements	Medium
Analytics Module	Graphs	Medium
	Charts	Medium

**Table 5.1 High Level Test Plan of the System**

## 5.4 System Test Cases

After determining the comprehensive test plan which is used to validate and verify functionalities of the system, the next step is to define the set of scenarios and conditions which should be tested under the defined test plan. Test cases are carefully defined by covering every aspect of the system. This section containing with test results only for Students Registration module and please refer the Appendix E for detailed actual test results with screen shots for other modules of the system.

### 5.4.1 Test Cases for Students Registration Module

The Table 5.2 shows the test cases and expected results defined for Students Registration module.

ID	Test Description	Steps for Testing	Expected Result
1	When New Students Registration form is loaded	Go to Students Registration option and click on 'New Students' option	"Date of Registration" and "Date of Birth" calendar should display the system date
2	When all the required fields are not filled with user input properly	Leave empty some required fields that marked with (*) character and click Add button	Display error message
3	When all the required fields are not filled with user input with correct pattern	Input some fields that does not match with the required input pattern that mentioned in the Help block area	Display error message
4	New students registration function	Input all the fields that match with correct input pattern and click "Add" button	Display message "New Student Added Successfully"
5	Search students function	Go to Students Registration option and click on 'Search Students' option	All active students should be loaded in to the Address book

6	View student's profile details	Go to Students Registration option and click on 'Edit Profile' option. Then select one student record and click 'View' button	All the input fields should be disabled and filled with relevant details of the student
7	Verify un-editable information in student's profile	Go to Students Registration option and click on 'Edit Profile' option. Then select one student record and click 'Edit' button	Un-editable information should be blocked for editing after loading the "UpdateStudentProfile" form
8	Edit some information in student's profile	Go to the same location as above and update some fields with correct pattern and click "Update Details" button	Display message "Student's Details Successfully Updated"
9	Inactivate selected student function	Go to Students Registration option and click on 'Edit Profile' option. Then select one student record and click 'Inactive' button	Display user warning message "Are you really sure to inactivate the selected student??"
10	Successfully inactivate a selected student	Do the same step as above and click 'Inactive' button. In the user warning message, click "Yes"	Display message "Student's Details Successfully Inactivated"
11	Abort the selected student inactive operation	Do the same step as above and click 'Inactive' button. In the user warning message, click "No"	System should abort the inactive operation and load the same page "Edit Students' Profile"

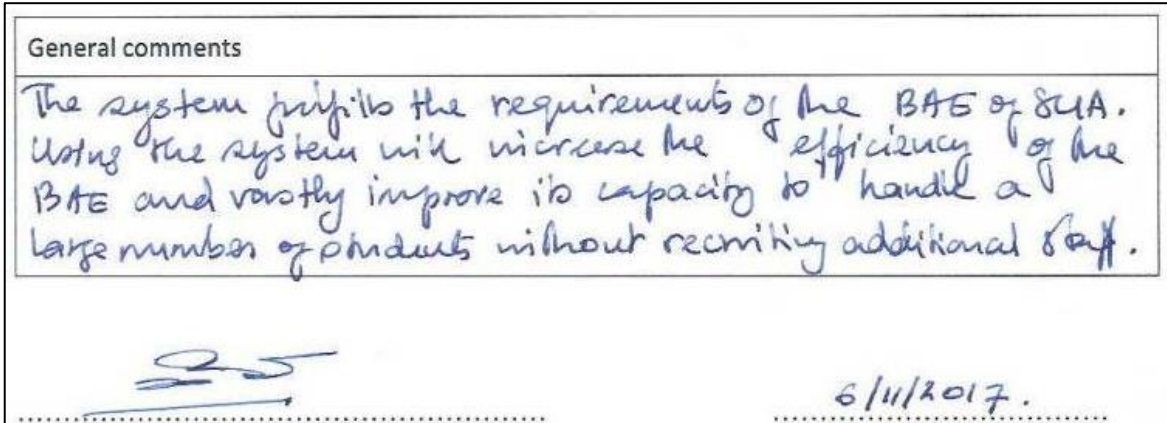
**Table 5.2 Students Registration Module Test Cases**



## 5.5 User Acceptance Testing

After completing the implementation work, the system was installed at the clients' location in order to go through the user acceptance testing process. The objective was to perform user acceptance testing with the help from potential users during a certain period of time. The user who has administrative privileges completed end to end testing of the system and some system defects were identified during the testing. Those identified defects were fixed and the fixed issues were successfully got verified by administrative user again. The course fee payment, counseling fee payment and monitoring due payment functionalities were tested by the potential users from accounts department of BAE. The functionalities dedicated for Student user level were successfully tested by few students from the institute.

At the final stage of the user acceptance testing, the comments and feedback about the new system were collected from users by presenting an evaluation questionnaire form. Some feedback comments received from a user is illustrated in Figure 5.1



General comments

The system fulfills the requirements of the BAE of SQA. Using the system will increase the efficiency of the BAE and vastly improve its capacity to handle a large number of products without recruiting additional staff.

*[Signature]* 6/11/2017.

**Figure 5.1 General Comments from a User**

User evaluation questionnaire form that was used for the survey and responses collected by a potential user has been included in “Appendix E – Test Results” section.

# CHAPTER 06: CONCLUSION

## 6.1 Introduction

Sri Lanka Institute of Architects is the governing body of the Architecture profession in Sri Lanka. A student can acquire the B.Arch degree in Architecture from University of Moratuwa or Higher Diploma in Architecture from the City School of Architecture in Colombo to gain exemptions from Parts I and II of SLIA; those with foreign qualifications will have to get their transcripts sent to SLIA and obtain the exemptions; thereafter they can register to undertake the SLIA Part III examination to acquire the Charter from the Sri Lanka Institute of Architects to practice individually or as a firm of architects as well as to perform institutionally. With the increasing number of students' registration, the institution has been facing a lot of difficulties during the past few years due to the manual processes that were established to conduct their daily operational activities.

## 6.2 Critical Assessment of the Project

The main objective of implementing the Students Management System for the SLIA was to provide a more organized and streamlined mechanism to manage the routine processes, those have been manually performed at BAE. The proposed solution was to implement a centralized web based system for SLIA that could help to streamline students' related operational activities in a more organized and efficient manner. In order to satisfy these objectives, the scope of the project was outlined according to following sub modules.

- Students' admission process handling system
- Students' payments handling and notification system
- Students' attendance handling and notification system
- Students' examination records handling system
- BAE course records handling system
- Staff information handling system
- Reports generating system

The specific customer requirements and the problem domain that expected to implement the web based solution have been properly identified in requirements analysis phase. Then the identified requirements were categorized as functional and non-functional requirements according to the defined scope of the project. After designing the solution, the implementation tasks were carried out module-wise successfully by following the activities and phases defined in Rational Unified Process methodology. After that the system testing was properly conducted by using wide range of test cases which cover the functionalities of the implemented system. The user acceptance testing was conducted using few real users and according to the feedback collected from the users, it can be mentioned that the objectives of the project have been fulfilled up to a satisfactory level and without major concerns raised by the client.

However due to the complexity and importance of the activities currently carried out by SLIA in the professional education field, the client organization would like to conduct a parallel run of the new Student Management System along with their existing manual system a period of time. This will help for a smooth transition to the new system as well as to identify and fix any new operational issues those might be encountered while using the new system for a long period of time.

### 6.3 Lessons Learnt During the Course of the Project

The final year project paved the path to learn really valuable lessons in my personal and professional life in IT industry. The project was a great opportunity to gain experience while carrying out all the activities in the Software Development Life Cycle that is developing a working software solution starting from the initial customer requirements. Following are rest of the notable lessons learnt during the course of the project.

- This was a great opportunity to improve my technical knowledge in the areas such as web development languages and database management techniques.
- Managing pressure and the stress while meeting various deadlines as well as when things are not going according to the initial plans was another good lesson.
- Working with a real client while participating for business communications with different people and mapping the discussions to achieve technical solution.

- How to plan and organize the work and how to carry out the tasks by following the given guidelines.
- Individual project was a very good opportunity to improve technical writing skills to a great extent.

It was a great challenge to discuss and negotiate about some of the requirements with the client since there were some requirements which were not technically feasible to implement during a course of an academic project.

## 6.4 Future Enhancements

Following are some of the suggestions that could be added as enhancements to the existing system to make the system a much improved product in future.

- **Blog and students discussion forum**

A blog or a students' discussion forum could be included as an enhancement module to cater as an online forum that facilitates the discussions between students, lecturers and counsellors of the institute.

- **Introduce credit card payment facility**

Currently the students' payments are accepted only through cash and an enhancement could be done in this area by introducing credit card payment facility in the system.

- **Improvements in the Reporting Module**

Good improvements could be done in the Reporting Module by introducing more management and administrative reports.

# REFERENCES

- [1] John H. Daniels Faculty of Architecture, Landscape, and Design. [Online]. Available: <https://www2.daniels.utoronto.ca/prospective-students/programs/bachelor-arts-architectural-studies> [Accessed: Jul 01, 2017].
- [2] D. Thakur, "Requirements Analysis in Software Engineering", Ecomputernotes.com, 2017. [Online]. Available: <http://ecomputernotes.com/software-engineering/requirementsanalysis>. [Accessed: Jul 24, 2017].
- [3] "Open Source Student Information System - Fedena", Projectfedena.org, 2017. [Online]. Available: <http://www.projectfedena.org/>. [Accessed: Jul 29, 2017].
- [4] "Fedena - School Management Software", Fedena.com, 2017. [Online]. Available: [https://www.fedena.com/feature\\_tour](https://www.fedena.com/feature_tour). [Accessed: Jul 29, 2017].
- [5] N. Morpus, "The Top 6 Free and Open Source School Administration Software - Capterra Blog", Blog.capterra.com, 2017. [Online]. Available: <http://blog.capterra.com/the-top-6-free-school-administration-software/>. [Accessed: Jul 31, 2017].
- [6] "Free School Software & Online School Management System", Fekara, 2017. [Online]. Available: <https://fekara.com/>. [Accessed: Jul 31, 2017].
- [7] r. singh, "10 Best School Management Software or School ERP System", Technofizi.net, 2017. [Online]. Available: <https://technofizi.net/best-school-management-software-or-school-erp-system/>. [Accessed: Sep 30, 2017].
- [8] Techopedia.com. 2017. What is a Functional Requirement? - Definition from Techopedia.[Online].Available: <https://www.techopedia.com/definition/19508/functional-requirement> [Accessed: Aug 02, 2017].
- [9] Eriksson, U. 2017. Functional Requirements vs Non Functional Requirements. [Online]. ReQtest. Available: <http://reqtest.com/requirements-blog/functional-vs-non-functional-requirements/> [Accessed: Aug 04, 2017].
- [10] Techterms.com. 2017. RUP (Rational Unified Process) Definition. [Online]. Available: <https://techterms.com/definition/rup> [Accessed: Aug 05, 2017].

- [11] "Definition of 'Systems Design' - The Economic Times", The Economic Times, 2017.[Online]. Available: <http://economictimes.indiatimes.com/definition/systems-design>. [Accessed: Aug 05, 2017].
- [12] "What is Object-Oriented Analysis and Design (OOAD)? - Definition from Techopedia", Techopedia.com, 2017. [Online]. Available: <https://www.techopedia.com/definition/21104/object-oriented-analysis-and-design-ooad>. [Accessed: Aug 11, 2017].
- [13] "Aqvatarius's profile on CodeCanyon", CodeCanyon, 2017. [Online]. Available: <https://themeforest.net/user/aqvatarius>. [Accessed: Oct 07, 2017].
- [14] "What is Software Testing? - Definition from Techopedia", Techopedia.com, 2017. [Online]. Available: <https://www.techopedia.com/definition/17681/software-testing>. [Accessed: Oct 07, 2017].
- [15] "Software Testing - Methods", www.tutorialspoint.com, 2017. [Online]. Available: [https://www.tutorialspoint.com/software\\_testing/software\\_testing\\_methods.htm](https://www.tutorialspoint.com/software_testing/software_testing_methods.htm). [Accessed: Oct 07, 2017].
- [16] "Software Testing Levels - Software Testing Fundamentals", Software Testing Fundamentals, 2017. [Online]. Available: <http://softwaretestingfundamentals.com/software-testing-levels/>. [Accessed: Oct 07, 2017].

# Appendix A - System Documentation

This technical documentation can be used as a set of instructions to complete the installation and configuration of the Students Management System. Additionally this documentation contains the Hardware and Software environments in which the web based system is expect to be installed.

## Hardware Requirements

The minimum hardware requirements have been explained in the Table A.1 below.

Hardware	Minimum Requirement
Processor	2.50 GHz Processor
Memory	2 GB RAM
Hard Disk	40 GB
Display	1366 * 768 Resolution compatible monitor recommended
Printer	Inkjet Printer or LaserJet Printer
Internet Connection	Minimum Speed (512Kbps) Broadband connection

**Table A.1 Hardware Requirements**

## Software Requirements

The minimum software requirements have been explained in the Table A.2 below.

Software	Minimum Requirement
Operating System	Windows 7, Windows 8, Windows 10
XAMPP	XAMPP Control Panel v3.2.1 or new version (Recommended)
Web Browser	Google Chrome (Version 61.0.3163.100) Firefox (Version 56.0)

**Table A.2 Software Requirements**

# Instructions for System Installation

## Step 1 - XAMPP Installation

- Download XAMPP for Windows from <https://www.apachefriends.org> (Refer Table A.2 for recommended version)
- Choose **C:\XAMPP** as installation path of the computer

## Step 2 -Install a Web Browser

- Install a web browser which is recommended in Table A.2

## Step 3 - System Code Extraction

- Open the CD and copy BAE folder and paste it to the directory path  
**C:\XAMPP\htdocs**

## Step 4 - MySQL Database Installation

- Open the web browser and go to the URL <http://localhost/phpmyadmin/>
- Create a new empty database by providing the name **db\_bae**. Navigate to the **Import** option and click **Choose file** button. Navigate in the CD and select **db\_bae.sql** file.
- Then click on the **GO** button that is located in the bottom of the page.

## Step 5 - Launching the System

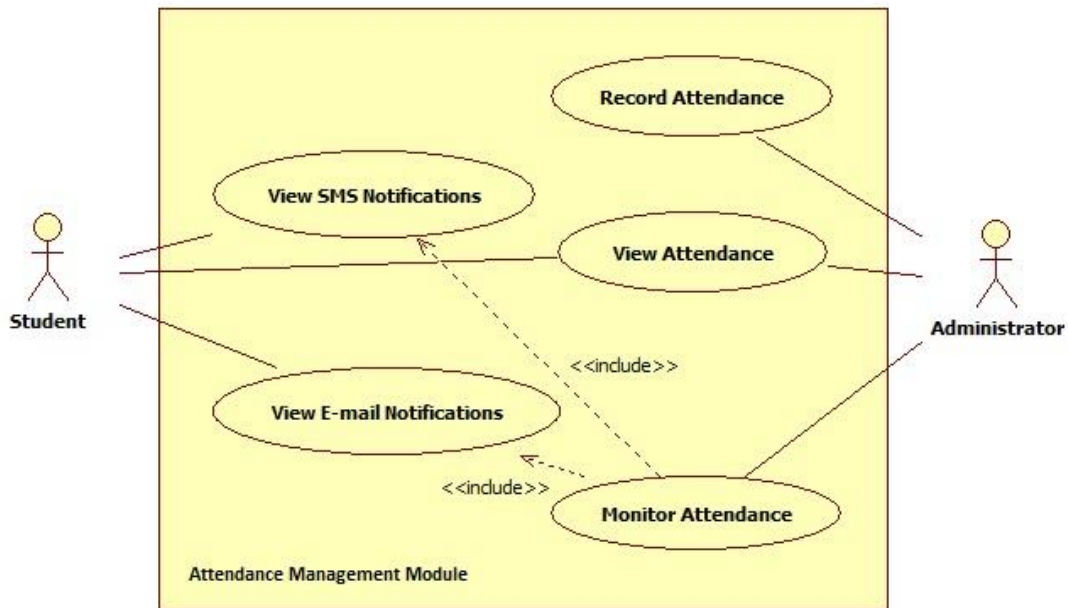
- Go to the notification area of the Windows Taskbar and open XAMPP control panel. Verify whether **Apache** and **MySQL** is running.
- Open the recommended web browser and type the URL: <http://localhost/BAE/> and press Enter button to launch the system.
- The initial login page of the Student Management System should appear at this stage.
- Operating instructions and more information of the system can be found in, Appendix C - User Documentation



# Appendix B - Design Documentation

The design documentation which has not been included in the Design Chapter (Chapter 3) is included in this appendix in order to have more in-depth knowledge about the design of the system for interested parties.

- **Use Case Diagram of the Attendance Management Module**



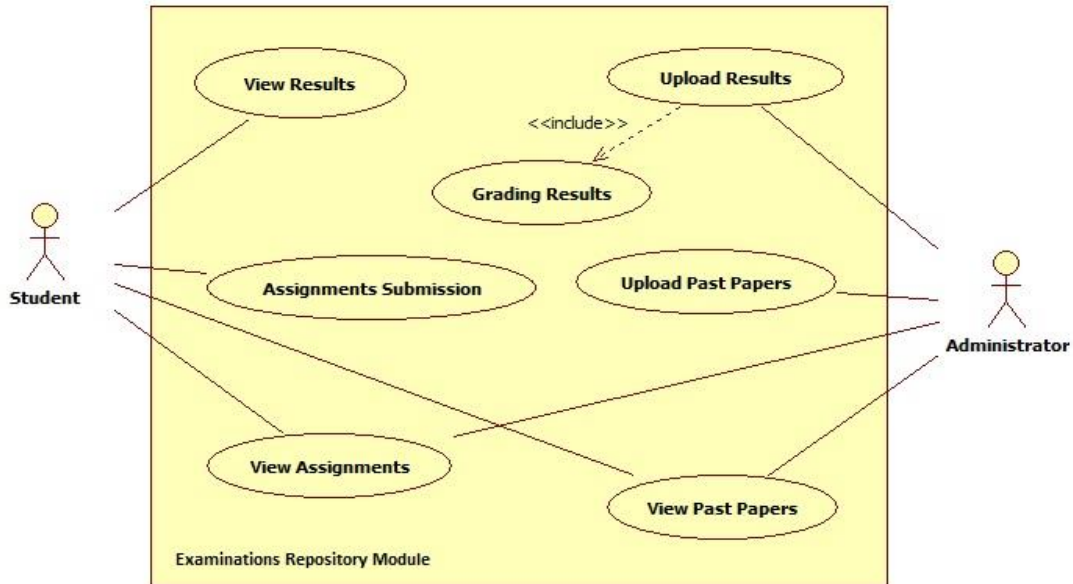
**Figure B.1 Use Case Diagram of Attendance Management Module**

<b>Use Case Name</b>	Attendance Management
<b>Actors</b>	Administrator, Student
<b>Description</b>	Administrator updates attendance records and performs attendance monitoring. Student and Administrator can view attendance records.
<b>Pre-conditions</b>	Course module wise attendance must be properly recorded for lecture dates in order to view and monitor attendance.
<b>Flow of Events</b>	<ol style="list-style-type: none"> <li>1. Attendance records for course modules are updated by Administrator.</li> <li>2. Recorded attendance can be viewed by Administrator and Students.</li> <li>3. Attendance monitoring carried out by Administrator.</li> <li>4. SMS and e-mail notifications will be sent for those students who</li> </ol>

	have not met minimum requirement for the course module.
Post-conditions	Attendance records should be properly maintained for generating accurate reports as well as attendance monitoring purpose.

**Table B.1 Use Case Description of Attendance Management Module**

- **Use Case Diagram of the Examination repository Module**



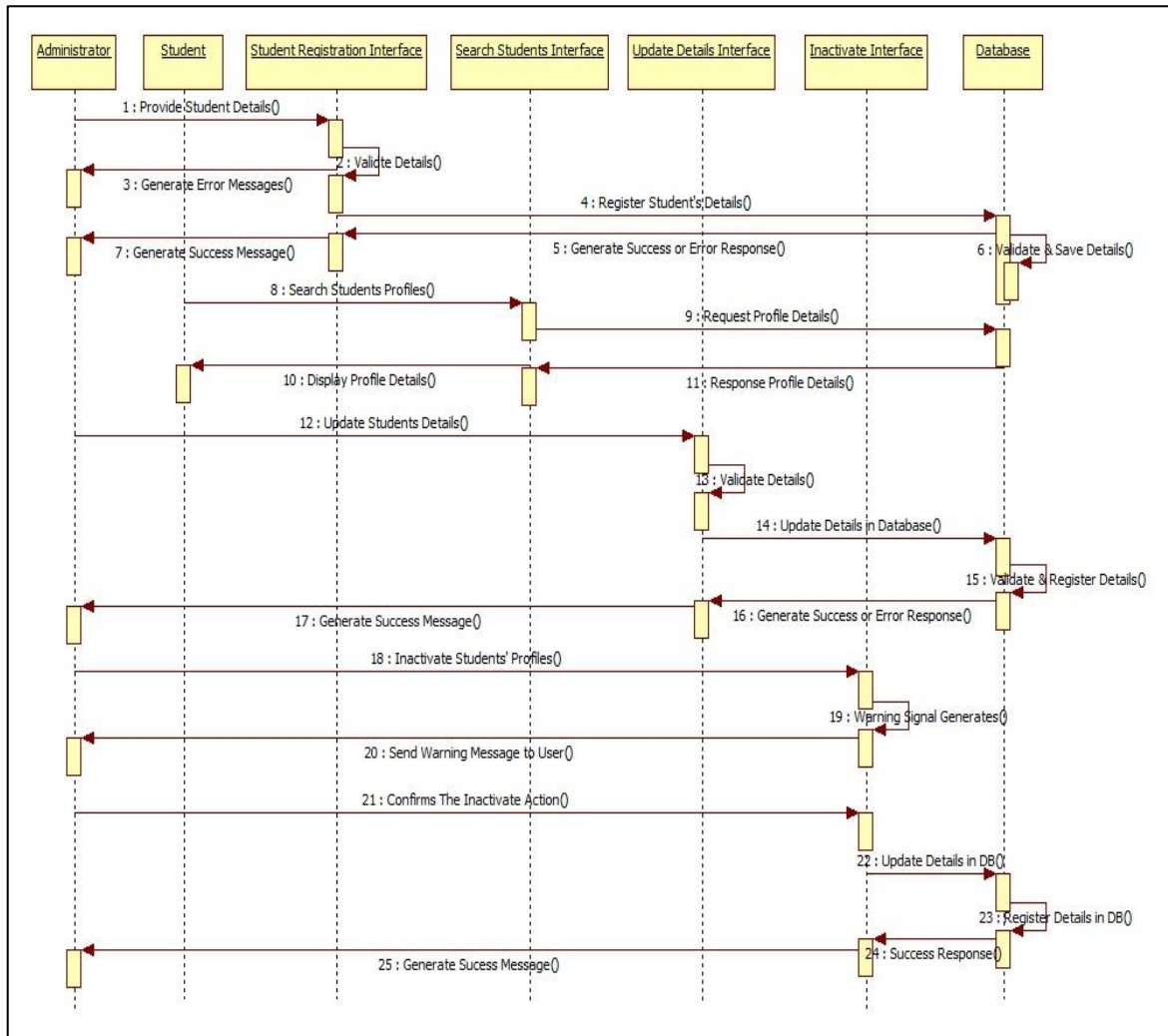
**Figure B.2 Use case Diagram of Examination Repository Module**

Use Case Name	Examination Functions Handling
Actors	Administrator, Student
Description	Various functions related to examinations are handled
Pre-conditions	Grades are calculated by system according to the marks that students have obtained for course modules.
Flow of Events	<ol style="list-style-type: none"> <li>1. Upload exam results and upload past papers functions are handled by administrator.</li> <li>2. View exam results, view past papers, assignments submission functions are performed by students.</li> <li>3. View assignments function is common for Administrator and Student</li> </ol>
Post-conditions	Examination results should get authorized by BAE before uploading to the system. Uploaded data should be accurate.

**Table B.2 Use Case Description of Examination Repository Module**

- **Sequence Diagram of the Students Registration Module**

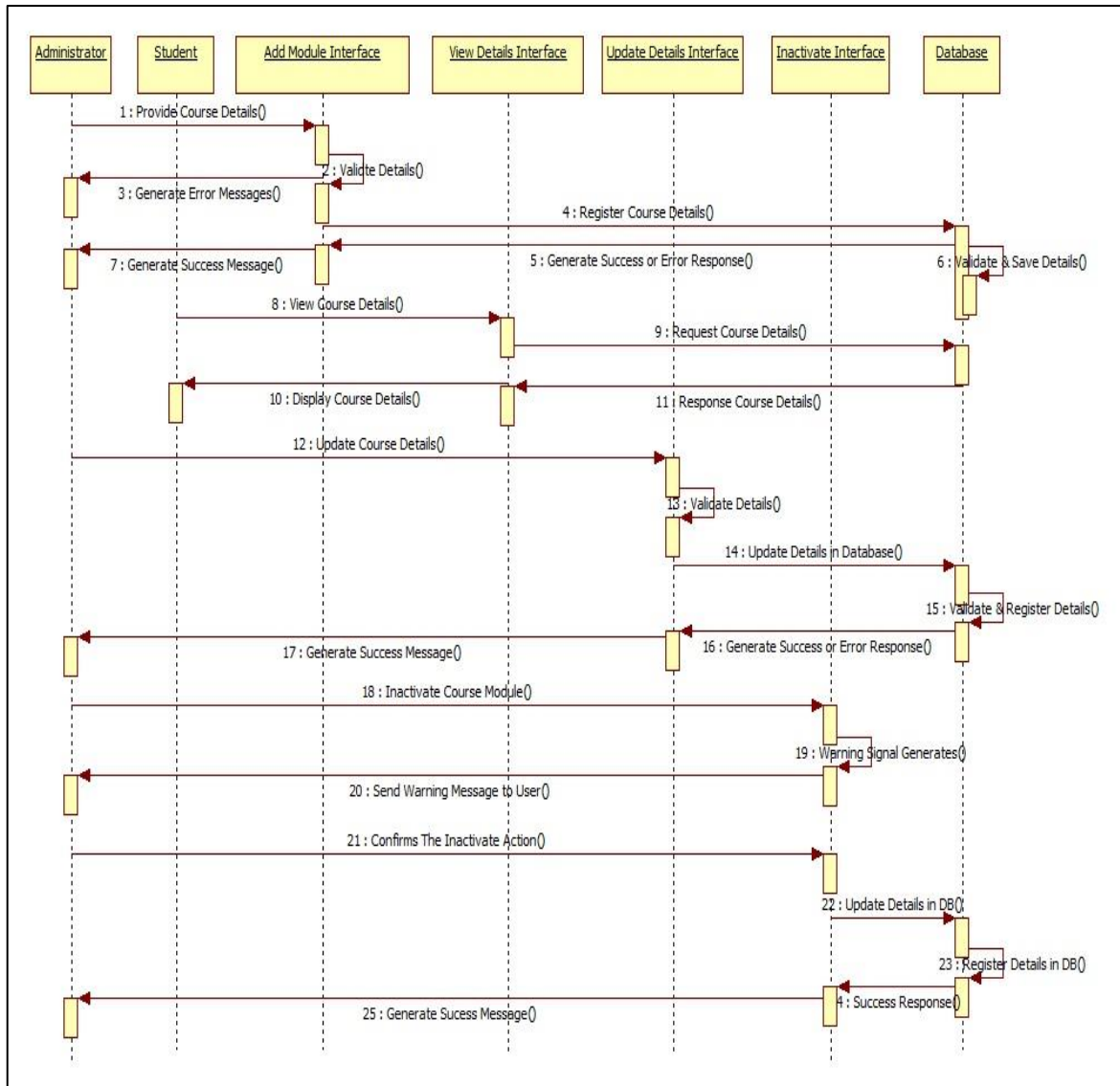
The sequence diagram created for the Students Registration Module is depicted in Figure B.3



**Figure B.3 Sequence Diagram of Students Registration Module**

- **Sequence Diagram of the Course Repository Module**

The sequence diagram created for the Course Repository Module is depicted in Figure B.4



**Figure B.4 Sequence Diagram of Course Repository Module**

# Appendix C - User Documentation

The purpose of this documentation is to provide instructions on how to productively make use of the system functionalities available in the Students Management System. The potential users of the system will be Students, Administrative staff and Accounts staff. Different access privileges have been granted for the system users based on the different roles of the system.

## Login page


Students Management System has been configured to access through the URL <http://localhost/BAE/>. Open a web browser that is recommended in the Table A.2 and navigate to the system using this URL. The initial login page of the system depicts in Figure C.1. System is accessible for all levels of the users through this initial login page by providing the NIC number as Username and with a valid Password.



Figure C.1 System Login Page

## Password Reset Process

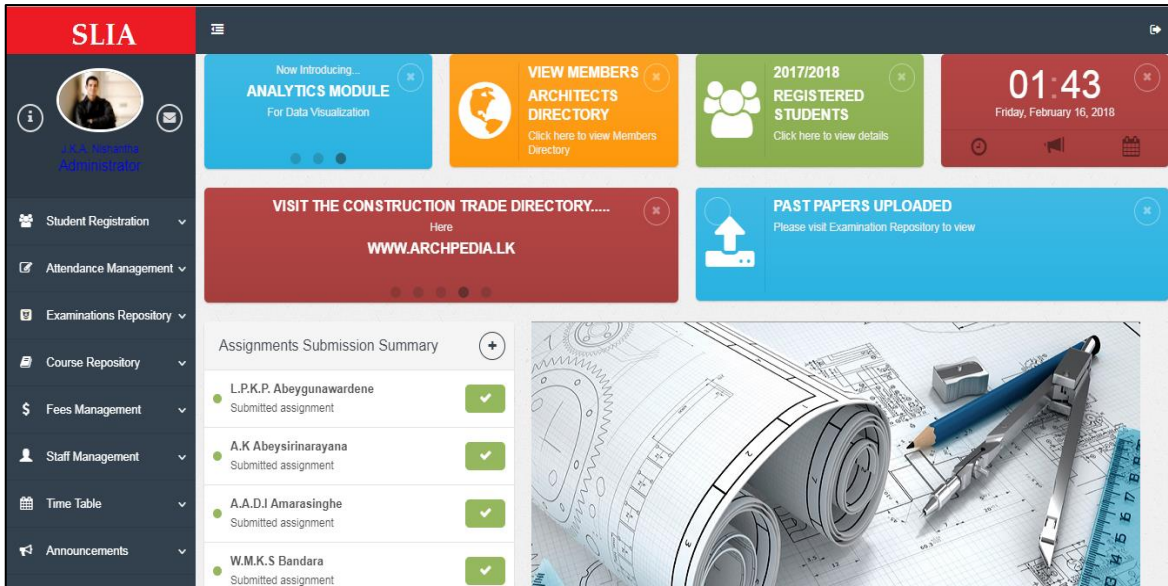
In the situations where system users have forgotten or lost their password, the password can be reset using the recovery page which can be accessed through the link **Forgot your password?** in the initial login page. The user should be able to prove the authenticity by providing mandatory information in order to complete the password reset process successfully. The password reset form is depicted in Figure C.2

The image shows a dark-themed user interface for a password reset process. At the top, there is a red rectangular box containing the text "SLIA" in white, serif font. Below this, the title "Reset Your Password" is displayed in a light gray font. The form consists of several input fields: "Username", "Password recovery question", "Your answer", "Enter New Password", and "Confirm Password". Each field is a dark gray rectangle with light gray placeholder text. At the bottom of the form, there are two buttons: a blue button labeled "Reset Password" and a green button labeled "Login Page".

**Figure C.2 Password Reset Window**

## Home Dashboard Page

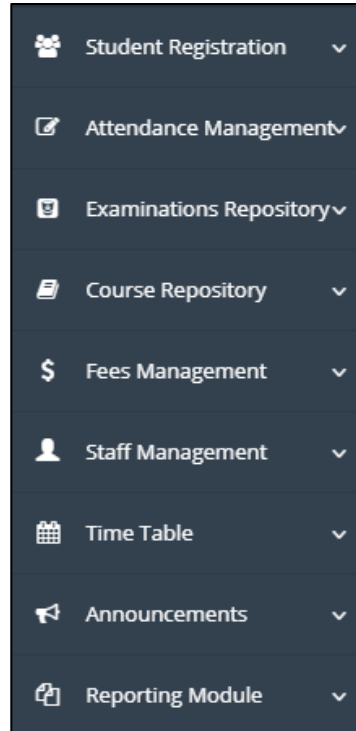
After completing a successful login attempt, the system users can access the Home Dashboard page which is depicted in Figure C.3. The Home Dashboard page is a common page which has been granted to be accessed by all levels of the users.



**Figure C.3 Home Dashboard Page**

## Navigation Menu for Module Structure

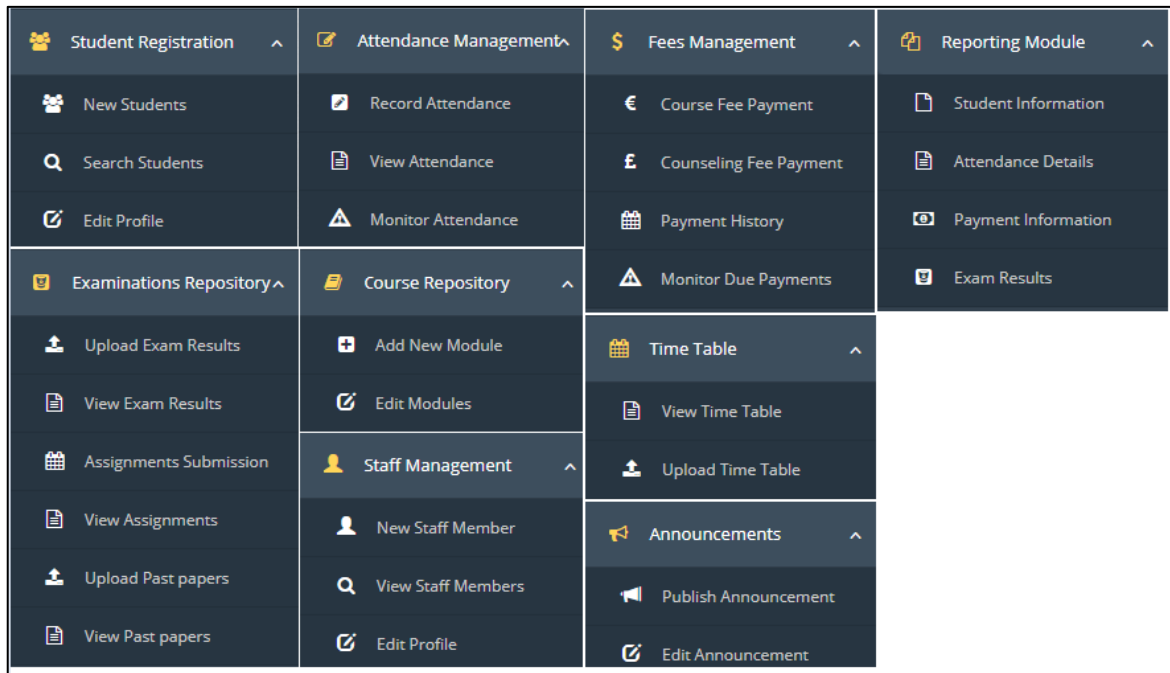
The Navigation Menu for module structure of the system is shown in Figure C.4.



**Figure C.4 Navigation Menu**


## Sub Menu Items of Main Modules

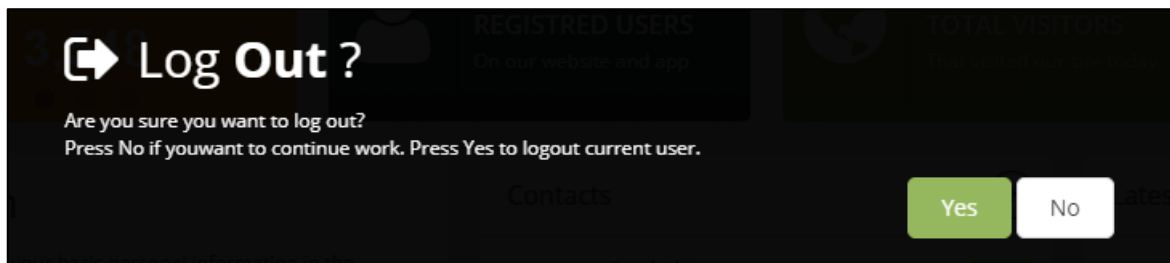
Sub menu items which are linked with co-functionalities of the main modules of the system is depicted in Figure C.5



**Figure C.5 Sub Menu Items under Main Modules**

## Logout from System

The logout feature of the system can be accessed through the icon  and this icon is located on the top right hand corner in Figure C.3. When the system logout feature was accessed by users, the verification message that is shown in Figure C.6 will be prompted in order to get the confirmation from users before login out from the system.

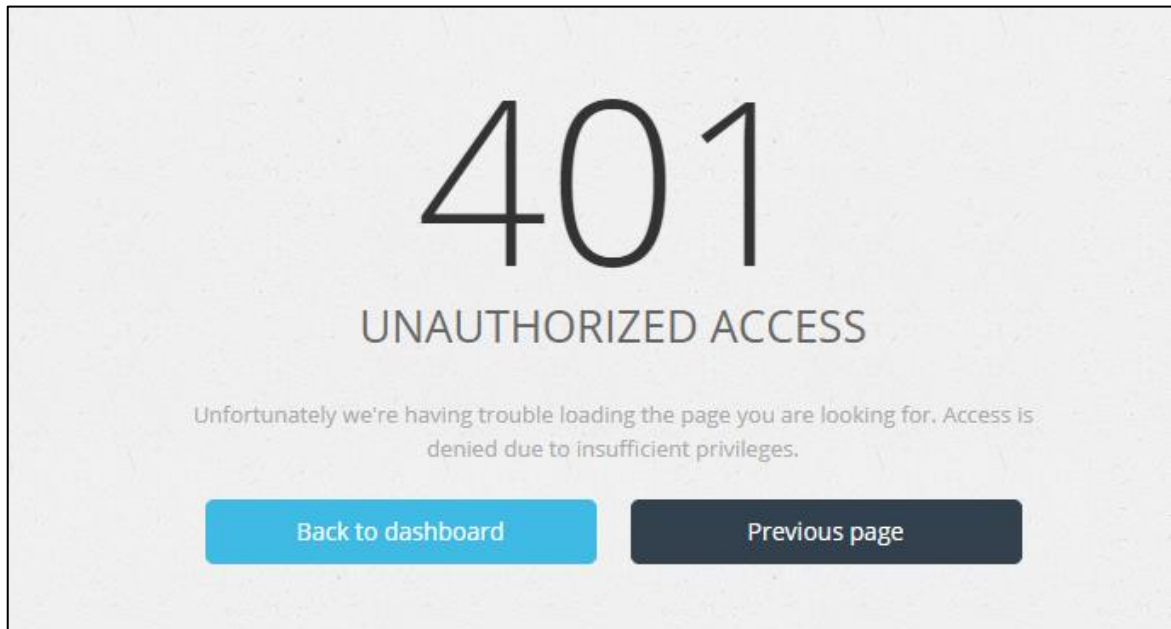


**Figure C.6 Verification Message before Logout**



## Insufficient Access Privileges Display Page

When a particular user attempts to perform a system function or attempts to access a menu item which have not configured to be accessed by the relevant user role, the unauthorized access error page depicted in Figure C.7 will be prompted to the relevant users.



**Figure C.7 Unauthorized Access Error Display Page**

## User Notification Message Types

User notification message types have been configured in the system to alert the users about the success or failure status of the operation which performed by users.

- **Success Message**

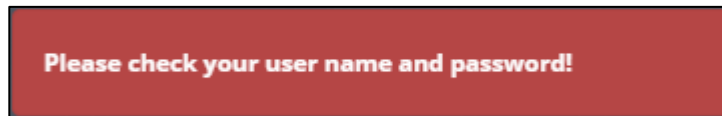
Figure C.8 depicts the success message that will be prompted to the users for a successful operation.



**Figure C.8 Success Message**

- **Failure Message**

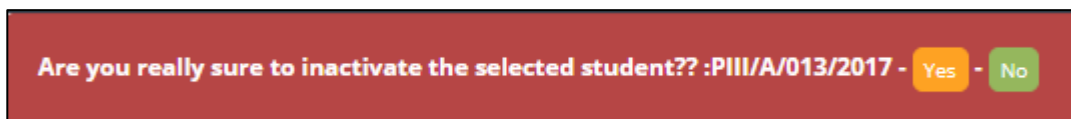
Figure C.9 depicts the failure message that will be prompted to the users for an unsuccessful operation.



**Figure C.9 Failure Message**

- **Warning Message**

Figure C.10 depicts the warning message that will be prompted to warn the users about the severity of the operation that the user attempts to perform.



**Figure C.10 Warning Message**

## Students Registration Module Student Registration ^

Functionalities such as new students registration, searching for registered students, view profile information of the students, update existing information and inactivate students' profiles can be performed through this module.

- **New Students Registration**

Access path: Student Registration → New Students

New students' registration process can be performed using this function. Only the Administrator of the system has access privileges for this function. All the required fields in this form should be filled-in according to the format that highlighted in help-block section to complete the registration process. Figure C.11 depicts an extract of the form.

**New Students Registration**

Registration NO\*  Date of Registration   
E.g. P1111A/012/2017 Select the Date of Registration

Name with Initials\*  Date of Birth   
A.B.C.Pereira Click on input field to select Date of Birth

**Figure C.11 New Students Registration Form**

- **Search Registered Students**

Access path: Student Registration → Search Students

Searching through the profiles of the students’ can be performed using this function. Figure C.12 depicts an extract of this feature.

**Search Students**

Use search to find contacts. You can search by: Name, Address or Phone.

L.P.K.P Abeygunawardene B.Arch UoM	A.K Abeysirinarayana B.Arch UoM	H.Y Amarakoon AIA(SL)	A.A.D.I Amarasinghe B.Arch UoM

**Figure C.12 Search Students Feature**

- **View, Update and Inactivate Operations**

Access path: Student Registration → Edit Profile

View the registered information and update / inactivate students’ profiles can be performed by the administrator using this function. The data-table shown in the Figure C.13 has been utilized in some of the sections in few other modules as well for performing view, update and inactivate operations where applicable. This data-table is equipped with powerful features such as real time data searching in the table and export records to other files, etc...

Students' Records Export Export Records Export Data

Show  entries Number of Entries Search Records Search:

Registration Number	Title	Name With Initials	Registration Date	NIC Number	Mobile Number	Action	Action	Action
PIII/A/001/2017	Ms.	L.P.K.P Abeygunawardene	2017-10-01	856123740V	0772753593			
PIII/A/002/2017	Ms.	A.K Abeyirinarayana	2017-10-01	865232330V	0716296542			
PIII/A/003/2017	Ms.	H.Y Amarakoon	2017-10-02	878160096V	0772443644	<span style="margin-left: 5px;">View Operations </span>		
PIII/A/004/2017	Mr.	A.A.D.I Amarasinghe	2017-10-02	873493569V	0772443644			
PIII/A/005/2017	Ms.	A.A Anam	2017-10-03	886272740V		<span style="margin-left: 20px;">Update Operations </span>		
PIII/A/006/2017	Ms.	F.R Arooz	2017-10-04	867732160V		<span style="margin-left: 20px;">Inactivate Operations </span>		

Showing 1 to 6 of 6 entries Previous 1 Next

**Figure C.13 View, Update, Inactivate Operations**

## Attendance Management Module

Functionalities such as record attendance, view attendance and monitor attendance can be performed through this module.

- **Record Attendance**

Access path: Attendance Management → Record Attendance

Only the system administrator has access privileges to record attendance of the students. Attendance should be recorded by selecting the relevant course module and the lecture date and loading them to the attendance table. The active students will be automatically loaded into the table and absence or presence for the lecture can be marked by selecting or deselecting the check-box. Figure C.14 describes the attendance recording procedure.

**Record Attendance**

Module Code:  Date of Lecture:

Select the Date of Lecture

Select the Module code from here Select the Date of Lecture

Select the Module code and Lecture Date and click load button

Registration Number	Title	Name With Initials	Module Code	Lecture Date	Attendance	Comment
PIII/A/001/2017	Ms.	L.P.K.P Abeygunawardene	ARP-01	2017-10-01	<input checked="" type="checkbox"/>	
PIII/A/002/2017	Ms.	A.K Abeyirinarayana	ARP-01	2017-10-01	<input checked="" type="checkbox"/>	
PIII/A/003/2017	Ms.	H.Y Amarakoon	ARP-01	2017-10-01	<input checked="" type="checkbox"/>	
PIII/A/004/2017	Mr.	A.A.D.I Amarasinghe	ARP-01	2017-10-01	<input checked="" type="checkbox"/>	
PIII/A/005/2017	Ms.	A.A Anam	ARP-01	2017-10-01	<input checked="" type="checkbox"/>	
PIII/A/006/2017	Ms.	F.R Arooz	ARP-01	2017-10-01	<input checked="" type="checkbox"/>	

Click to Record attendance for the selected module and date    Mark Attendance using this check box

**Figure C.14 Record Attendance Procedure**

- **View Attendance**

Access path: Attendance Management → View Attendance

Attendance can be viewed by selecting the relevant course module and the lecture date and clicking the ‘View Attendance’ button as explained in Figure C.15.

The screenshot shows the 'View Attendance' interface. At the top, there are two input fields: 'Module Code' with a dropdown menu labeled 'Select Module' and 'Date of Lecture' with a date picker set to '2017-10-26'. Below these fields are buttons for 'Clear', 'View Attendance', and 'Refresh'. A table below displays attendance records with columns for 'Registration Number', 'Module Code', 'Lecture Date', 'Attendance', and 'Comment'. The table contains six rows of data. The first row shows a student with registration number P1111/A/001/2017, module code ARP-01, and lecture date 2017-10-01, with a checked attendance box and a comment 'This student has present for the Course Module'. The second row shows a student with registration number P1111/A/002/2017, module code ARP-01, and lecture date 2017-10-01, with a checked attendance box. The third row shows a student with registration number P1111/A/003/2017, module code ARP-01, and lecture date 2017-10-01, with a checked attendance box. The fourth row shows a student with registration number P1111/A/004/2017, module code ARP-01, and lecture date 2017-10-01, with a checked attendance box. The fifth row shows a student with registration number P1111/A/005/2017, module code ARP-01, and lecture date 2017-10-01, with an unchecked attendance box and a comment 'This student is absent for the lecture'. The sixth row shows a student with registration number P1111/A/006/2017, module code ARP-01, and lecture date 2017-10-01, with an unchecked attendance box. At the bottom of the interface are buttons for 'Record Attendance' and 'Monitor Attendance'.

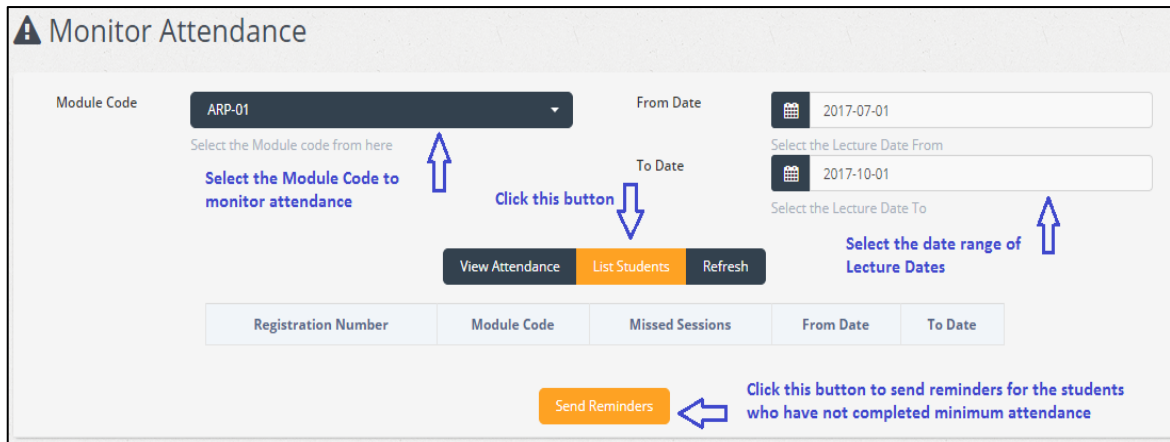
Registration Number	Module Code	Lecture Date	Attendance	Comment
P1111/A/001/2017	ARP-01	2017-10-01	<input checked="" type="checkbox"/>	This student has present for the Course Module
P1111/A/002/2017	ARP-01	2017-10-01	<input checked="" type="checkbox"/>	
P1111/A/003/2017	ARP-01	2017-10-01	<input checked="" type="checkbox"/>	
P1111/A/004/2017	ARP-01	2017-10-01	<input checked="" type="checkbox"/>	
P1111/A/005/2017	ARP-01	2017-10-01	<input type="checkbox"/>	This student is absent for the lecture
P1111/A/006/2017	ARP-01	2017-10-01	<input type="checkbox"/>	

**Figure C.15 View Attendance Procedure**

- **Monitor Attendance**

Access path: Attendance Management → Monitor Attendance

Only the system administrator has access privileges to monitor attendance of the students. The purpose of this functionality is to identify the students who have not achieved the required minimum number of sessions for a particular course module during a period of time and send reminders for those students to maximize their participation for the lectures. Figure C.16 illustrates the steps which should be followed in monitoring attendance.



**Figure C.16 Attendance Monitoring Procedure**

## Examination Repository Module Examinations Repository ^

This module contains with the functionalities such as upload exam results and view exam results, assignments submission and view submitted assignments and upload past papers and view past papers.

- **Upload Exam Results**

Access path: Examination Repository → Upload Exam Results

Uploading exam results can only be performed by the administrator and following are the steps which should be followed to complete this process successfully.

- 1) Navigate to the 'Upload Exam Results' form using the given access path.
- 2) Select Registration Number from drop down list.
- 3) Select results Effective Date from the calendar object.
- 4) Click 'List Modules' button to load the course modules list.
- 5) Feed exam results of the student in the 'Marks' column of the table.
- 6) Click 'Upload Results' button to upload the results for the selected student.

Figure C.17 illustrates these steps in more detail.

**Upload Exam Results** ← Step 1

Step 2 ↓

Step 3 ↓

Registration NO: Select Registration Number  
Select the Registration Number from here

Effective Date: 2017-10-27  
Select the Effective Date

Clear List Modules ← Step 4

Year	Registration Number	Module Code	Marks	Comment	Effective Date	Approved By
2017	PIII/A/001/2017	ARP-01	88		2017-10-01	BAE Approved ▼
2017	PIII/A/001/2017	LPR-02	92		2017-10-01	BAE Approved ▼
2017	PIII/A/001/2017	MDS-03	77	Optional field	2017-10-01	BAE Approved ▼
2017	PIII/A/001/2017	PRA-04	53		2017-10-01	BAE Approved ▼
2017	PIII/A/001/2017	RES-05	69		2017-10-01	BAE Approved ▼

↑ Automatically loaded to the table, can be changed if required.

Clear Upload Results Refresh

**Figure C.17 Upload Exam Results Procedure**

- **View Exam Results**

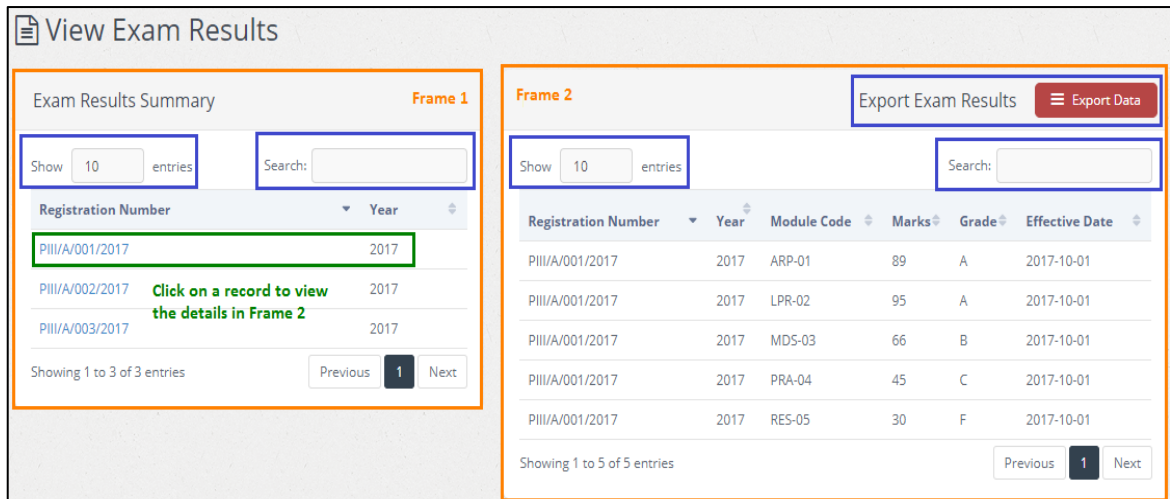
Access path: Examination Repository → View Exam Results

Viewing exam results functionality is allowed to be accessed by both students and administrators. The user interfaces which are designed for the purpose are self-explanatory and contain useful search functions as well. The form contains with two frames as follows.

- 1) Frame 1 - Contains a summary with Registration Number and Year
- 2) Frame 2 - Contains exam results in detail with marks and grades obtained.

When a user initially loads the page, form loads only the ‘Frame 1’, that is exam result summary page. When the user needs to view exam results in detail, it’s required to click on the Registration Number link and the exam results details are shown in ‘Frame 2’

Figure C.18 illustrates the exam results viewing procedure in more detail.



**Figure C.18 View Exam Results Procedure**

## Fees Management Module \$ Fees Management

The accounts department of the BAE has privileges to access this module. Functionalities such as course fee payment, counseling fee payment, check payment history and monitoring due payments consist in this module.

- **Course Fee Payment**

Access path: Fees Management → Course Fee Payment

Following are the steps which should be followed to complete course fee payment process successfully.

- 1) Navigate to the 'Course Fee Payment' form using the given access path.
- 2) Select Registration Number from drop down list.
- 3) Check the balance payment of the student by clicking Chk bal button. The balance payment will be loaded to the Balance 35000  
PIII/A/001/2017 text box and appropriate installment number also will be loaded to the Installment Number 2 text box.
- 4) Manually provide the Receipt Number.
- 5) Select Date of Payment from the calendar object.
- 6) Provide payment amount in the text box.



- 7) Click 'Add' button to add the course fee payment for the selected student.
- 8) Following success message will appear and click 'Print' button to generate and print the payment invoice.

New Course Payment Added Successfully: Registration Number PIII/A/001/2017. Do you want to print the payment Invoice?
Print

Figure C.19 illustrates these steps in more detail and Figure C.20 depicts the payment invoice that was generated through the system for the successful payment.

The screenshot shows a web interface for 'Course Fee Payment'. It includes several input fields and buttons:

- Step 1:** Registration NO dropdown menu with 'PIII/A/001/2017 - L.P.K.P Abeygunawardene' selected.
- Step 2:** Balance field showing 'PIII/A/001/2017' circled in orange, with a 'Check balance payment' button and 'Chk bal' button.
- Step 3:** A 'Print' button (not explicitly labeled in the screenshot but implied by the text).
- Step 4:** Receipt Number\* input field with '0123456' entered.
- Step 5:** Date of Payment calendar icon with '2017-10-01' selected.
- Step 6:** Amount\* input field with '15000' entered.
- Step 7:** Installment Number input field with '1' entered, circled in orange.

At the bottom, there are 'Clear', 'Add', and 'Refresh' buttons.

**Figure C.19 Course Fee Payment Procedure**

The invoice is from the Sri Lanka Institute of Architects (SLIA BAE). It contains the following information:

**Sri Lanka Institute of Architects**  
 Board Of Architectural Education Sri Lanka  
 SLIA BAE

No. 120/7, Vidya Mawatha  
 Colombo 07  
 Sri Lanka

Receipt Number #: 0123456  
 Payment Date : 2017-10-01 00:00:00  
 Payment Type : Course Fees

Payment Method	Cash
Amount	15000
Payment Details	
Student's Registration Number	PIII/A/001/2017
Installment Number	1
Thank you for the course fee payment of 15000 Rupees	
<b>Total: Rs.15000</b>	

**Figure C.20 Course Fee Payment Invoice**

- **Counseling Fee Payment**

Access path: Fees Management → Counseling Fee Payment

The users can follow the same guidelines provided in the course fee payment procedure with some minor changes which applicable for the counseling fee payment method. However the user interface design changes in counseling fee payment method are self-explanatory. Figure C.21 illustrates the counseling fee payment procedure.

The screenshot shows a web form titled "Counseling Fee Payment". It contains several input fields: "Registration NO" with a dropdown menu showing "PIII/A/001/2017 - L.P.K.P Abeygunawardene"; "Receipt Number\*" with the value "012346"; "Amount\*" with the value "7500"; "Date of Payment" with a calendar icon and the date "2017-10-27"; and "Year" with the value "1". A blue circle highlights the "Year" field. To the right of the "Date of Payment" field, there is a "Balance" section with a "Check balance payment" button and a "Chk bal" button. Below the "Date of Payment" field, there is a note: "You can check the remaining payment from here". At the bottom of the form, there are three buttons: "Clear", "Add", and "Refresh". A blue note at the bottom right says: "Users have to check the 'Fee Year' in Counseling Fee Payment method".

**Figure C.21 Counseling Fee Payment Procedure**

## Course Repository Module Course Repository

The purpose of this module is to maintain information of the course modules conducted by BAE. Functionalities such as add new course module, view information of already available modules, update information whenever necessary and inactivate course modules when such a requirement exists.

- **Add New Module**

Access path: Course Repository → Add New Module

Adding new course module function can only be performed by the administrator and following are the steps which should be followed to complete this process successfully.

- 1) Navigate to the 'Add New Module' form using the given access path.
- 2) Provide Module Code according to the correct naming convention.
- 3) Follow the Step 3, Step 4, Step 5, Step 6, Step 7 and Step 8 according to the guidelines provided in Figure C.22

- 4) The information required to be provided in Step 9 and Step 10 will be helpful to monitor attendance of the students, so provide those details correctly.
- 5) Click **Add** button to save this information.

Figure C.22 illustrates this process in more detail.

The screenshot shows a form titled 'Add New Module' with the following fields and steps:

- Step 1:** Module Code\* (PRA-04)
- Step 2:** Course Description\* (Practice of Architecture)
- Step 3:** Description of the course module
- Step 4:** Course Title\* (PRACTICE)
- Step 5:** Coordinator\* (Archt. Jayantha Perera)
- Step 6:** Mode of Testing\* (Viva/QP/Assignment)
- Step 7:** Lecturer\* (Ar. Prof. Chitra Weddikara)
- Step 8:** Examiner\* (Archt. Janaka Dharmasena)
- Step 9:** No. of Lectures (8)
- Step 10:** Min. Lectures (6)
- Step 11:** Clear and Add buttons

**Figure C.22 Add New Module Procedure**

- **Edit Modules**

Access path: Course Repository → Edit Modules

View, update and inactivate operations for the course module details can be performed using this function. Figure C.23 illustrates these functions in more detail.

The screenshot shows the 'Edit Course Modules' interface with the following table:

Module Code	Course Title	Coordinator	Lecturer	Examiner	Testing Mode	No. of Lectures	Min. Lectures	Action	Action	Action
ARP-01	PROFESSION	Archt. Rukshan Widyalanckara	Ar. Prof. Chitra Weddikara	Archt. Asoka Widanapathirana	Viva	6	4	View	Edit	Inactive
LPR-02	LEGAL PROCESSES	Archt. Prasanna Silva	Ms. Kanthi De Silva	Mr. Ashan Stanislaus	Assignment	6	4	View	Edit	Inactive
MDS-03	MANAGEMENT & DELIVERY OF SERVICES	Archt. Vasantha De Silva	Ar. Prof. Chitra Weddikara	Archt. D B Navaratne	PCPA/QP/Assignment	8	6	View	Edit	Inactive
PRA-04	PRACTICE	Archt. Jayantha Perera	Mr. Wasantha Meewaddana	Archt. Janaka Dharmasena	Viva/Assignment	8	6	View	Edit	Inactive
RES-05	PROJECT RESEARCH	Archt. Veranjan Kurukulasuriya	Archt. Veranjan Kurukulasuriya	Archt. Janaka Dharmasena	Report	6	4	View	Edit	Inactive

**Figure C.23 View, Update and Inactivate Functions**

## Reporting Module

Generating various types of reports to cater reporting requirements of different levels of users of the system is the purpose of this module.

- **Payment Information**

Access path: Reporting Module → Payment Information

Payment information reports are one type of report among the various reports that can be generated through the system. Figure C.24 illustrates about this report type in detail.

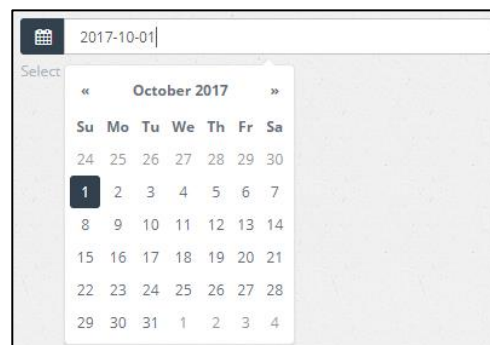


Payment Summary Reports	
Report 1	Course Fee Payment Report
Report 2	Counseling Fee Payment Report
Report 3	Due Payments Report

**Figure C.24 Payment Summary Reports**

- **Calendar Object**

The purpose of the calendar object is to assist the user in selecting particular date or date range when performing a certain function through the system. This object has been used in different areas in Reporting module and several other modules as well. Calendar object is shown in Figure C.25.



**Figure C.25 Calendar Object**

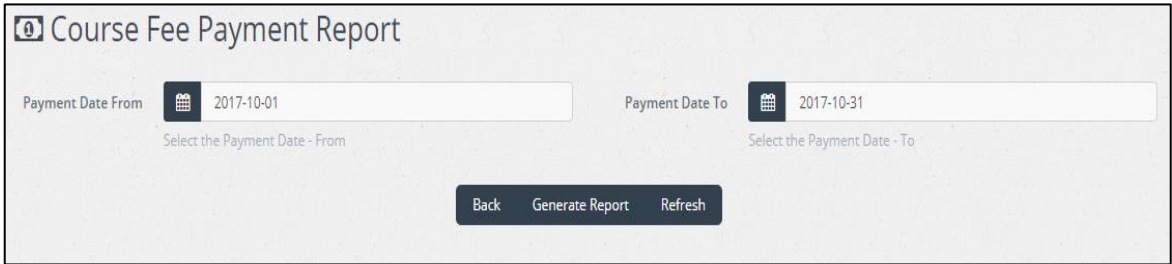
# Appendix D - Management Reports

Generating reports for management purposes is one of the most useful features of the Students Management System. This section describes about few of the reports that could be generated using reporting module.

## Payment Summary Reports

- **Course Fee Payment Report**

Course Fee Payment Report can be generated as a daily report, monthly report or quarterly report by specifying the relevant date range using the calendar object according to the preference of the management team. Figure D.1 illustrates the method of generating this report and Figure D.2 shows the Course Fee Payment Report which was generated as a monthly report.



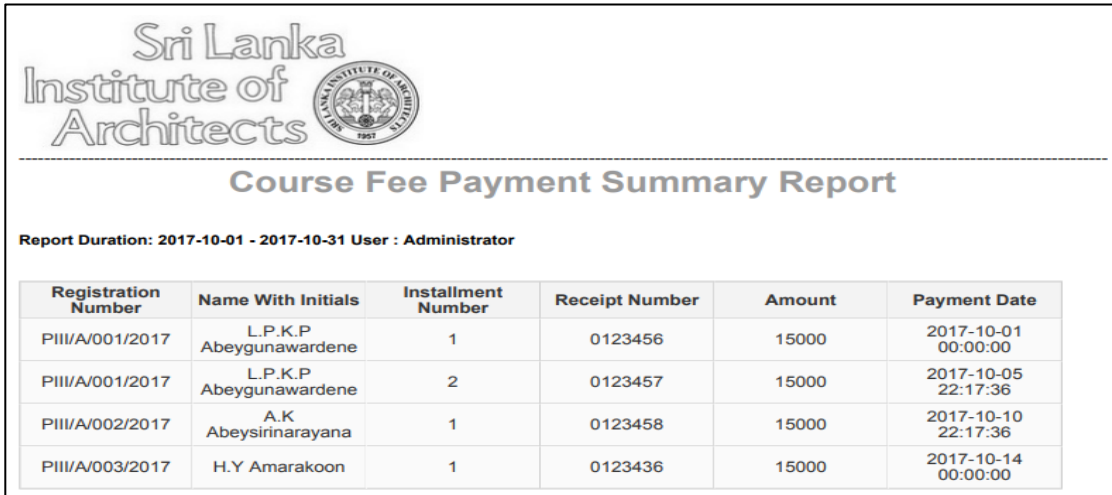
Course Fee Payment Report

Payment Date From  Payment Date To

Select the Payment Date - From Select the Payment Date - To

Back Generate Report Refresh

**Figure D.1 Course Fee Payment Report Interface**



Sri Lanka  
Institute of  
Architects

**Course Fee Payment Summary Report**

Report Duration: 2017-10-01 - 2017-10-31 User : Administrator

Registration Number	Name With Initials	Installment Number	Receipt Number	Amount	Payment Date
PIII/A/001/2017	L.P.K.P Abeygunawardene	1	0123456	15000	2017-10-01 00:00:00
PIII/A/001/2017	L.P.K.P Abeygunawardene	2	0123457	15000	2017-10-05 22:17:36
PIII/A/002/2017	A.K Abeyririnarayana	1	0123458	15000	2017-10-10 22:17:36
PIII/A/003/2017	H.Y Amarakoon	1	0123436	15000	2017-10-14 00:00:00

**Figure D.2 Course Fee Payment Monthly Report**

- **Counseling Fee Payment Report**

Counseling Fee Payment Report also can be generated as a daily report, weekly report or quarterly report according to the preference of the management team. Figure D.3 illustrates the method of generating this report and Figure D.4 shows the Counseling Fee Payment Report which was generated as a weekly report.

**Figure D.3 Counseling Fee Payment Report Interface**

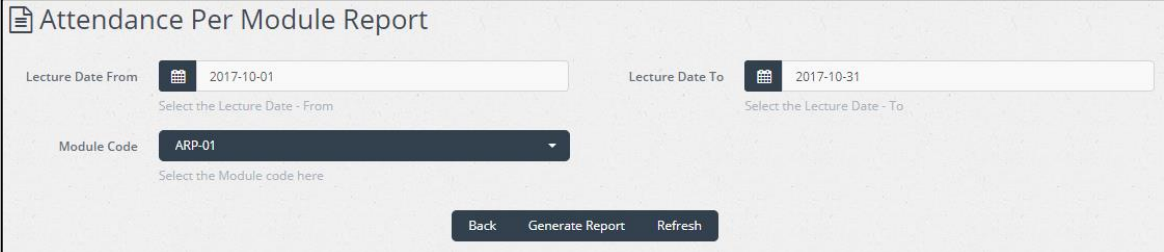
Registration Number	Name With Initials	Fee_Year	Receipt Number	Amount	Payment Date
PIII/A/001/2017	L.P.K.P Abeygunawardene	1	0123456	7500	2017-10-01 00:00:00
PIII/A/001/2017	L.P.K.P Abeygunawardene	2	0123454	7500	2017-10-04 00:00:00
PIII/A/002/2017	A.K Abeysirinarayana	1	0123457	7500	2017-10-01 00:00:00
PIII/A/003/2017	H.Y Amarakoon	1	0123458	7500	2017-10-03 00:00:00
PIII/A/004/2017	A.A.D.I Amarasinghe	1	0123459	7500	2017-10-03 00:00:00
PIII/A/005/2017	A.A Anam	1	0123455	7500	2017-10-04 00:00:00

**Figure D.4 Counseling Fee Payment Weekly Report**

## Attendance Summary Reports

- **Attendance per Module Report**

Attendance per Module Report which can be generated as a weekly or monthly report by summarizing attendance of the students provides very useful information for management staff. Figure D.5 illustrates the method of generating this report and Figure D.6 shows the Attendance per Module Report which was generated as a monthly report.

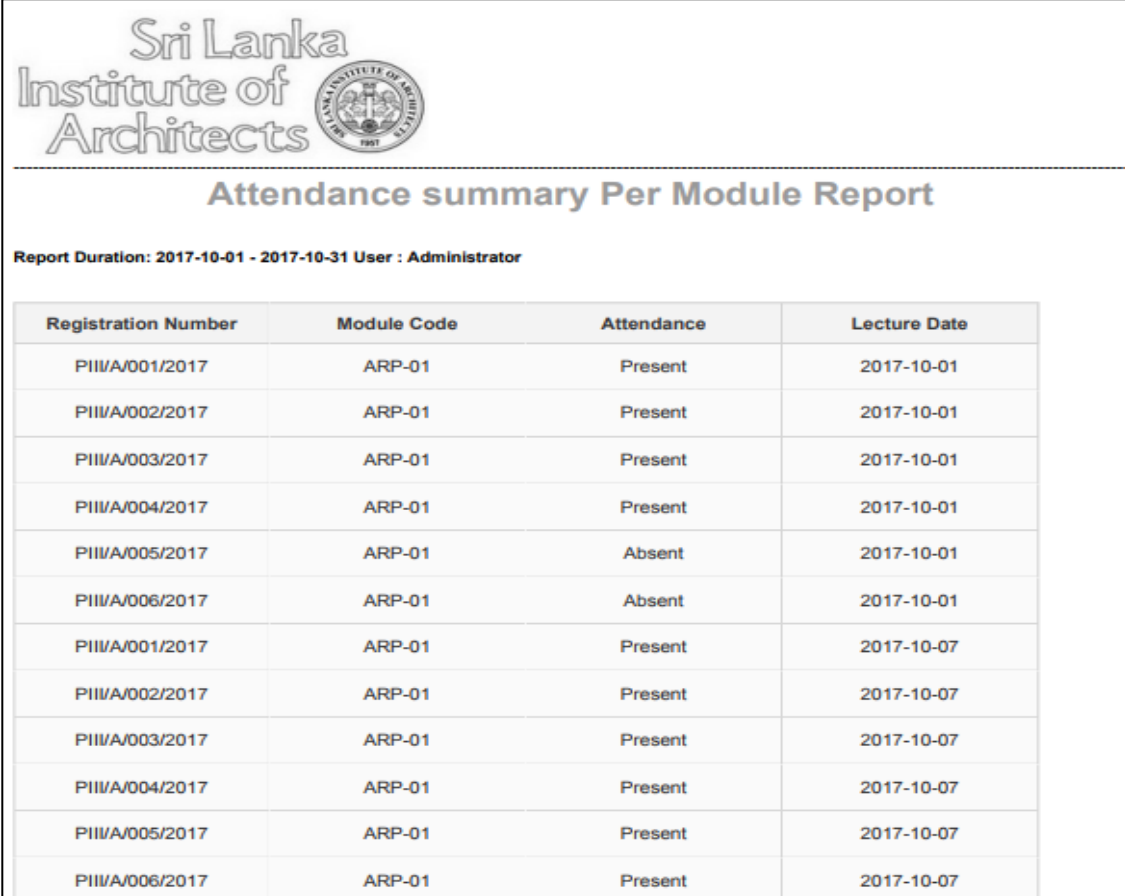


Attendance Per Module Report

Lecture Date From: 2017-10-01  
Lecture Date To: 2017-10-31  
Module Code: ARP-01

Back Generate Report Refresh

**Figure D.5 Attendance per Module Report Interface**



Sri Lanka Institute of Architects

**Attendance summary Per Module Report**

Report Duration: 2017-10-01 - 2017-10-31 User : Administrator

Registration Number	Module Code	Attendance	Lecture Date
PIII/A/001/2017	ARP-01	Present	2017-10-01
PIII/A/002/2017	ARP-01	Present	2017-10-01
PIII/A/003/2017	ARP-01	Present	2017-10-01
PIII/A/004/2017	ARP-01	Present	2017-10-01
PIII/A/005/2017	ARP-01	Absent	2017-10-01
PIII/A/006/2017	ARP-01	Absent	2017-10-01
PIII/A/001/2017	ARP-01	Present	2017-10-07
PIII/A/002/2017	ARP-01	Present	2017-10-07
PIII/A/003/2017	ARP-01	Present	2017-10-07
PIII/A/004/2017	ARP-01	Present	2017-10-07
PIII/A/005/2017	ARP-01	Present	2017-10-07
PIII/A/006/2017	ARP-01	Present	2017-10-07

**Figure D.6 Attendance per Module Monthly Report**

## Students' Information Reports

These types of reports contain profile details of students' which can be generated as a weekly, monthly or quarterly report by specifying the registration date of the student. Figure D.7 illustrates the method of generating this report and Figure D.8 shows the Students' Information Report which was generated as a monthly report.

Active / Inactive Students Information

Reg.Date From: 2017-10-01  
Select the Date of Registration - From

Reg.Date To: 2017-10-31  
Select the Date of Registration - To

Student Status\*: Active  
Select Student's Status From Here

Back Generate Report Refresh

**Figure D.7 Students' Information Report Interface**

Sri Lanka Institute of Architects

**Student Information Report**


Report Duration: 2017-10-01 - 2017-10-31 User : Administrator

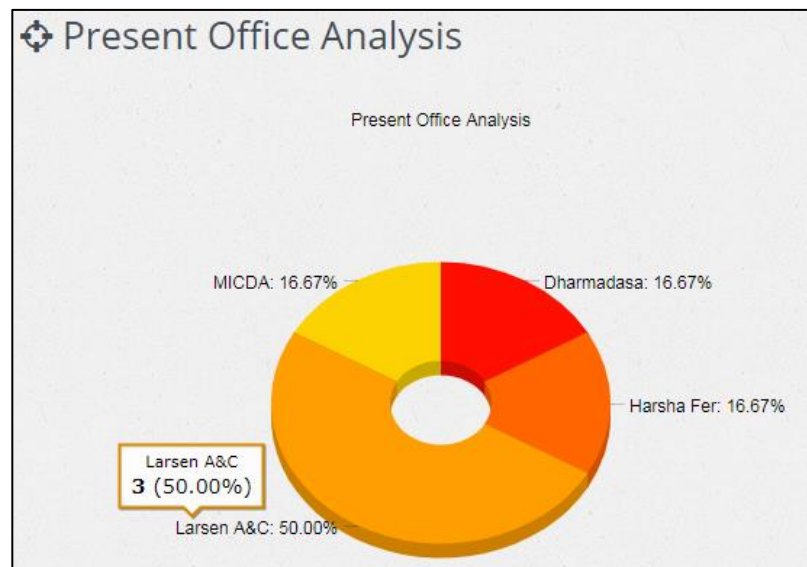
Registration Number	Registration Date	Name With Initials	NIC Number	Gender	Present Office	Current Status
PIII/A/001/2017	2017-10-01	L.P.K.P Abeygunawardene	856123740V	Female	Larsen A&CE Pvt Ltd	Active
PIII/A/002/2017	2017-10-01	A.K Abeysirinarayana	865232330V	Female	Larsen A&CE Pvt Ltd	Active
PIII/A/003/2017	2017-10-02	H.Y Amarakoon	878160096V	Female	MICDA	Active
PIII/A/004/2017	2017-10-02	A.A.D.I Amarasinghe	873493569V	Male	Larsen A&CE Pvt Ltd	Active
PIII/A/005/2017	2017-10-03	A.A Anam	886272740V	Female	Dharmadasa & Wijayarathne Associates	Active
PIII/A/006/2017	2017-10-04	F.R Arooz	867732160V	Female	Harsha Fernando Associates	Active

**Figure D.8 Student Information Monthly Report**

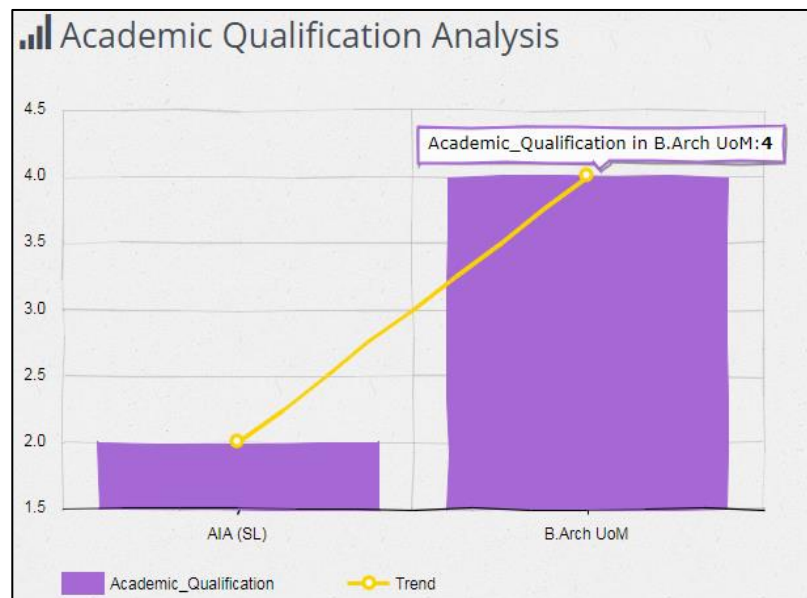


## Graphs and Charts for Decision Making

Data visualization is a great concept that helps for making decisions in an organization. Creating graphs and charts by visualizing existing data can be performed through the Analytics module  Analytics of the system. Figure D.9 illustrates a pie chart which was created by visualizing 'Present Office Analysis' information and Figure D.10 illustrates a bar chart which was created by visualizing 'Academic Qualifications' of the registered students.



**Figure D.9 A Pie Chart Generated through System**






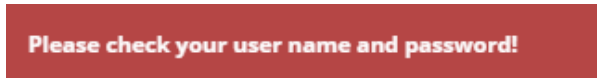
**Figure D.10 A Bar Chart Generated through System**

# Appendix E - Test Results

This chapter consists of the major test cases along with the actual test results to show that all aspects of the system have been verified and specification has been met.

## Test Cases for User Login Module



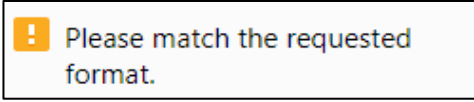
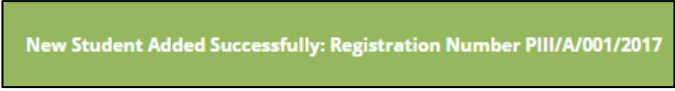
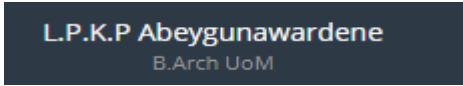


Table E.1 contains the test cases executed for user login module and actual test results captured during system evaluation.


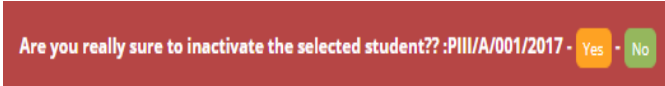

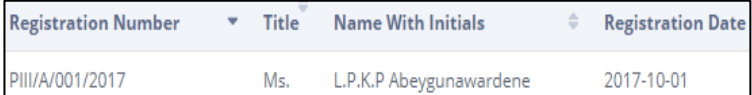
Test No.	Steps for Testing	Expected Result and Actual Result	Status
1	Input correct username Input correct password Click - Log In button	Successfully login in to the system 	Pass
2	Input correct username Input incorrect password Click - Log In button	Prompt "Check your user name or password" 	Pass
3	Input incorrect username Input correct password Click - Log In button	Prompt "Check your user name or password" 	Pass
4	Input incorrect username Input incorrect password Click - Log In button	Prompt "Check your user name or password" 	Pass

**Table E.1 Test Results of User Login Module**

## Test Cases for Students Registration Module

Table E.2 contains the test cases executed for students' registration module and actual test results captured during tests.

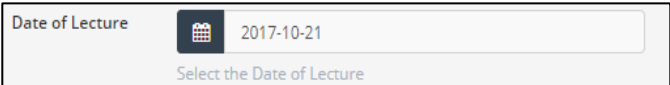
Test No.	Steps for Testing	Expected Result and Actual Result	Status
1	Go to Students Registration option and click on 'New Students' option	"Date of Registration" and "Date of Birth" calendar should display the system date 	Pass
2	Leave empty some required fields that marked with (*) character and click Add button	Display error message 	Pass
3	Input some fields that does not match with the required input pattern that mentioned in the Help block area	Display error message 	Pass
4	Input all the fields that match with correct input pattern and click "Add" button	Display message "New Student Added Successfully" 	Pass
5	Go to Students Registration option and click 'Search Students' option	All active students should be loaded in to the Address book area 	Pass
6	Go to Students Registration option and click on 'Edit Profile' option. Then select one student record and click 'View' button	All the input fields should be read only and filled with relevant details of the student. 	Pass
7	Go to Students Registration option and click on 'Edit Profile' option. Then select one student record and click 'Edit' button	Un-editable information should be blocked for editing after loading the "UpdateStudentProfile" form 	Pass

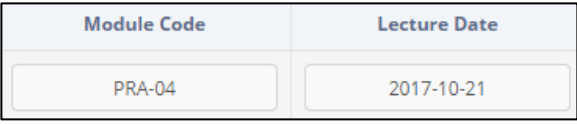
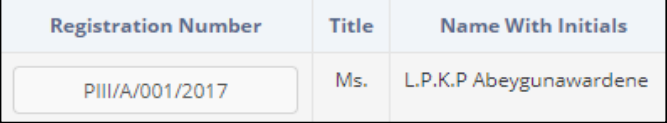
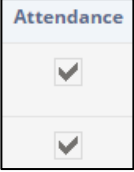
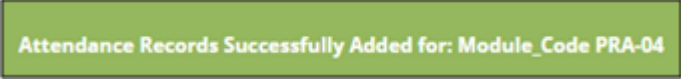
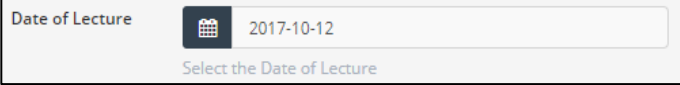
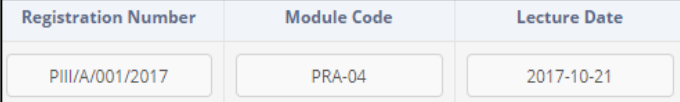
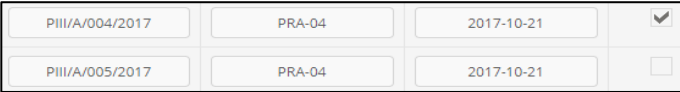
8	Go to the same location as above and update some fields with correct pattern and click "Update Details" button	Display message "Student's Details Successfully Updated" 	Pass
9	Go to Students Registration option and click on 'Edit Profile' option. Then select one student record and click 'Inactive' button	Display user warning message "Are you really sure to inactivate the selected student??" 	Pass
10	Do the same step as above and click 'Inactive' button. In the user warning message, click "Yes"	Display message "Student's Details Successfully Inactivated" 	Pass
11	Do the same step as above and click 'Inactive' button. In the user warning message, click "No"	System should abort the inactivate operation and load the same page "Edit Students' Profile" → Pass 	

**Table E.2 Test Results of Students Registration Module**

## Test Cases for Attendance Management Module

Table E.3 contains the test cases executed for attendance management module and actual test results captured during tests.




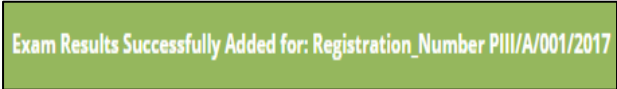
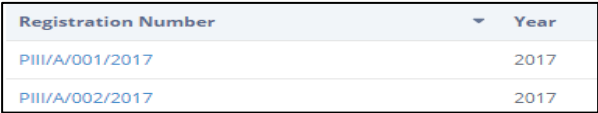
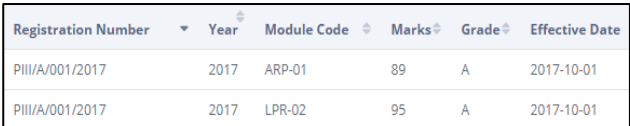
Test No.	Steps for Testing	Expected Result and Actual Result	Status
1	Go to Attendance Management option and click on Record Attendance	"Date of Lecture" calendar should be filled with the system date 	Pass

2	Select a Module Code from drop down and the Date of Lecture and click "Load" button	Selected Module Code and the Date of Lecture should be populated to the attendance uploading table 	Pass
3	Select a Module Code from drop down and the Date of Lecture and click "Load" button	With the same result of above step, all active students should be loaded to the same table 	Pass
4	Select a Module Code from drop down and the Date of Lecture and click "Load" button	With the same result of above step, Attendance check box that loaded to the table should be marked as checked for all students 	Pass
5	After verifying the results of above test cases, click "Record Attendance" button	Display message "Attendance Records Successfully Added" 	Pass
6	Go to Attendance Management option and click on View Attendance	"Date of Lecture" calendar should be filled with the system date 	Pass
7	Select the Module Code and the Date of Lecture that attendance recorded and click "View Attendance" button	Selected Module Code, Date of Lecture, all active students should be populated to the view table 	Pass
8	Do the same step as in the above test case	For absent students the check box should be unchecked whereas present students it should be checked. 	Pass

**Table E.3 Test Results of Attendance Management Module**

## Test Cases for Examination Repository Module

Table E.4 contains the test cases executed for examination repository module and actual test results captured during tests.

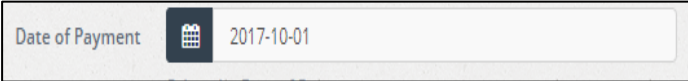
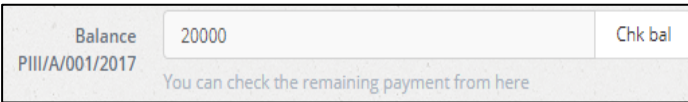
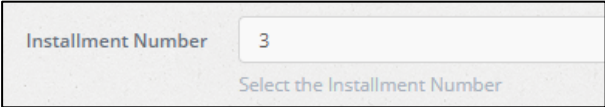

Test No.	Steps for Testing	Expected Result and Actual Result	Status
1	Go to Examinations Repository option and click Upload Exam Results	"Effective Date" calendar should be filled with the system date 	Pass
2	Select a Registration Number and the results effective date and click "List Modules" button	Registration Number and the results effective date should be loaded to the results table 	Pass
3	Follow the same steps as of the above test case	All course modules and the current year should be loaded to the results table 	Pass
4	Follow the same steps as of the above test case and mention the marks and click "Upload Results" button	Display message "Exam Results Successfully Added" 	Pass
5	Go to Examinations Repository option and click on View Exam Results	Exam Results Summary page should be filled with Registration numbers and Results year (Panel – 1) 	Pass
6	Click on the link of one Registration Number to view the detailed exam results	Detailed results should be visible in panel -2 of the result view page 	Pass

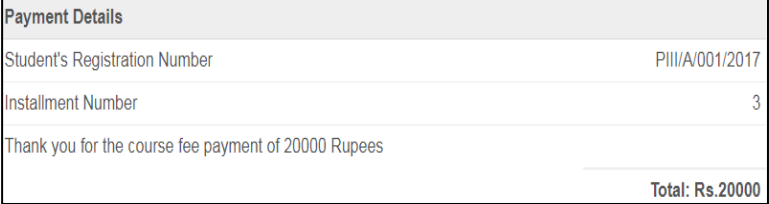
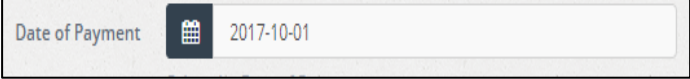
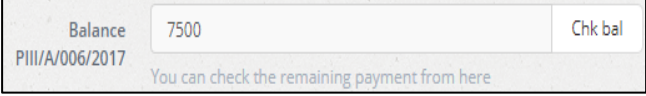
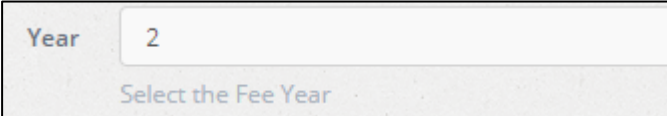
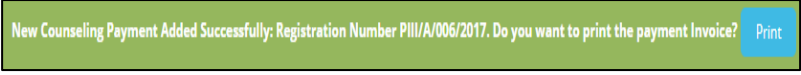
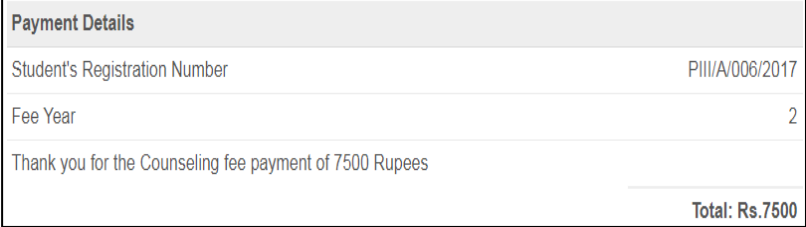
7	Verify the grades of the students against the marks obtained for course modules	<p>Grades should be properly calculated according to the pre-defined grades calculating logic</p> <table border="1" data-bbox="670 264 1330 443"> <thead> <tr> <th>Registration Number</th> <th>Year</th> <th>Module Code</th> <th>Marks</th> <th>Grade</th> <th>Effective Date</th> </tr> </thead> <tbody> <tr> <td>P111/A/001/2017</td> <td>2017</td> <td>ARP-01</td> <td>89</td> <td>A</td> <td>2017-10-01</td> </tr> <tr> <td>P111/A/001/2017</td> <td>2017</td> <td>LPR-02</td> <td>95</td> <td>A</td> <td>2017-10-01</td> </tr> <tr> <td>P111/A/001/2017</td> <td>2017</td> <td>MDS-03</td> <td>66</td> <td>B</td> <td>2017-10-01</td> </tr> <tr> <td>P111/A/001/2017</td> <td>2017</td> <td>PRA-04</td> <td>45</td> <td>C</td> <td>2017-10-01</td> </tr> </tbody> </table>	Registration Number	Year	Module Code	Marks	Grade	Effective Date	P111/A/001/2017	2017	ARP-01	89	A	2017-10-01	P111/A/001/2017	2017	LPR-02	95	A	2017-10-01	P111/A/001/2017	2017	MDS-03	66	B	2017-10-01	P111/A/001/2017	2017	PRA-04	45	C	2017-10-01	Pass
Registration Number	Year	Module Code	Marks	Grade	Effective Date																												
P111/A/001/2017	2017	ARP-01	89	A	2017-10-01																												
P111/A/001/2017	2017	LPR-02	95	A	2017-10-01																												
P111/A/001/2017	2017	MDS-03	66	B	2017-10-01																												
P111/A/001/2017	2017	PRA-04	45	C	2017-10-01																												

**Table E.4 Test Results of Examination Repository Module**

## Test Cases for Fees Management Module

Table E.5 contains the test cases and actual test results for fees management module.

Test No.	Steps for Testing	Expected Result and Actual Result	Status
1	Go to Fees Management option and click Course Fee Payment	<p>"Date of Payment" calendar should be filled with the system date</p> 	Pass
2	Select a Registration Number from drop down and check the balance payment by clicking "Chk Bal" button	<p>Balance payment should be loaded to the "Balance" text box along with the selected Registration Number</p> 	Pass
3	Do the same steps as of the above test case	<p>The Installment Number text box should be filled with the next correct Installment Number</p> 	Pass
4	Fill the Receipt Number and Payment amount and click "Add" button	<p>Display message "New Course Payment Added Successfully"</p> <p>Pass</p> 	



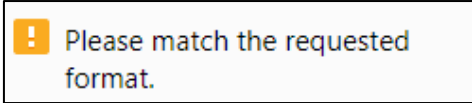

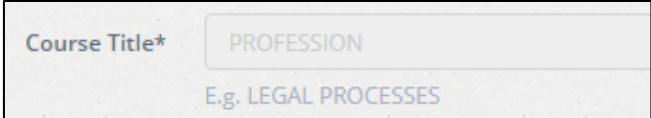
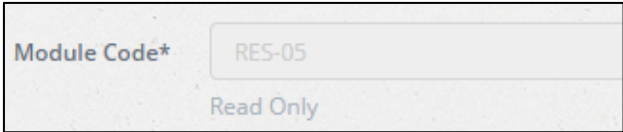
5	Do the same steps as of the above test case and click the "Print" button on the successful message for payment invoice verification	<p>Payment invoice should be generated → Pass</p> 	
6	Go to Fees Management option and click Counseling Fee Payment	<p>"Date of Payment" calendar should be filled with the system date</p> 	Pass
7	Select a Registration Number from drop down and check the balance payment by clicking "Chk Bal" button	<p>Balance payment should be loaded to the "Balance" text box along with the selected Registration Number</p> 	Pass
8	Do the same steps as of the above test case	<p>The Year text box should be filled with the next applicable payment Year</p> 	Pass
9	Fill the Receipt Number and Payment amount and click "Add" button	<p>Display message "New Counseling Payment Added Successfully"</p> <p style="text-align: center;">Pass</p> 	
10	Do the same steps as of the above test case and click the "Print" button on the successful message for payment invoice verification	<p>Payment invoice should be generated → Pass</p> 	

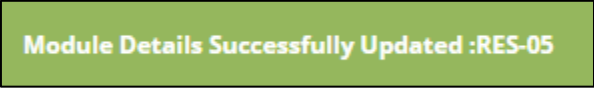
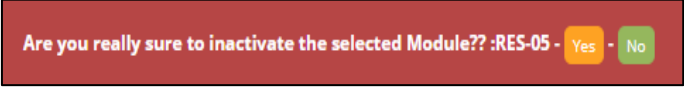

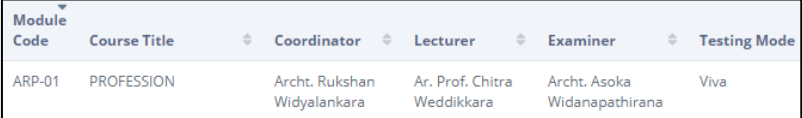
**Table E.5 Test Results of Fees Management Module**



## Test Cases for Course Repository Module

Table E.6 contains the test cases executed for course repository module and actual test results captured during tests.

Test No.	Steps for Testing	Expected Result and Actual Result	Status
1	Go to Course Repository option and click on Add New Module	All the input fields must be empty except No.of Lectures and Min.Lectures fields populated as 1 	Pass
2	Leave empty some required fields that marked with (*) character and click Add button	Display error message 	Pass
3	Input some fields that does not match with the required input pattern that mentioned in the Help block area	Display error message 	Pass
4	Input all the fields that match with correct input pattern and click "Add" button	Display message "New Module Added Successfully" 	Pass
5	Go to Course Repository option and click on 'Edit Modules' option. Then select one module record and click 'View' button	All the input fields should be read only and filled with relevant course details 	Pass
6	Go to Course Repository option and click on 'Edit Modules' option. Then select one module record and click 'Edit' button	Un-editable information should be blocked for editing after loading the "UpdateModuleDetails" form 	Pass

7	Go to the same location as above and update some fields with correct pattern and click "Update Details" button	Display message "Module Details Successfully Updated" 	Pass
8	Go to Course Repository option and click on 'Edit Modules' option. Then select one module record and click 'Inactive' button	Display user warning message "Are you really sure to inactivate the selected Module??" 	Pass
9	Do the same step as above and click 'Inactive' button. In the user warning message, click "Yes"	Display message "Module Details Successfully Inactivated" 	Pass
10	Do the same step as above and click 'Inactive' button. In the user warning message, click "No"	System should abort the inactivate operation and load the same page "Edit Course Module" → Pass 	

**Table E.6 Test Results of Course Repository Module**

# User Evaluation Questionnaire Form

A user evaluation questionnaire form completed by a potential user is shown in Figure E.1

User Evaluation Questionnaire – Students’ Management System

Name of the user : Arih. Veraniya Kurukulasinga

Designation : Senior Vice President / SLIA

No:	Evaluating Criteria	Very Good	Good	Average	Poor	Very Poor
1	Overall design of the system	✓				
2	Ease of learning functions	✓				
3	Ease of using features	✓				
4	Ease of training new users	✓				
5	Access privileges on different users	✓				
6	Understandability of error messages		✓			
7	Navigation menu structure design		✓			
8	Attractiveness of the system		✓			
9	Reporting formats	✓				
10	User friendliness	✓				

General comments

The system fulfills the requirements of the BAE of SLIA. Using the system will increase the efficiency of the BAE and vastly improve its capacity to handle a large number of products without recruiting additional staff.

Signature: [Signature] Date: 6/11/2017.

**Senior Vice President**  
Sri Lanka Institute of Architects

**Senior Vice President**  
Sri Lanka Institute of Architects

Figure E.1 User Evaluation Questionnaire Form

# Appendix F - Code Listing

Major code segments of the implemented functionalities have been included in this document. Please refer the CD-ROM for complete set of code listing of all modules of the Students Management System.

- **Code Segment for Recording Students' Attendance**

```
<?php

require_once '../..//control/config.php';

try {
    if (isset($_POST['Add'])) { //Record attendance by capturing the data
posting from attendance table in view form
        foreach ($_POST["Module_Code"] as $rec => $value) {
            $Module_Code = $_POST["Module_Code"][$rec];
            $Lecture_Date = $_POST["Lecture_Date"][$rec];
            $Registration_Number = $_POST["Registration_Number"][$rec];
            if ($_POST["Attendance"][$rec] == "") {
                $Attendance = 'Absent';
            } else {
                $Attendance = 'Present';
            }
            $Comment = $_POST["Comment"][$rec];
            $sql = 'INSERT INTO
tbl_students_attendance (Module_Code, Lecture_Date, Registration_Num
ber, Attendance, Comment) '

            'VALUES (:Module_Code, :Lecture_Date, :Registration_Number, :Attendan
ce, :Comment) ' ;

            $stmt = $conn->prepare($sql);
            $stmt->execute(array('Module_Code' => $Module_Code,
'Lecture_Date' => $Lecture_Date,
'Registration_Number' => $Registration_Number,
'Attendance' => $Attendance,
'Comment' => $Comment));
        }
        $conn = null;

        $_SESSION['SUCESSS'][] = 'Attendance Records Successfully Added for:
Module_Code ' . $Module_Code;
    }
}
```

- **Code Segment for New Students Registration**

```

<?php

require_once '../..../control/config.php';

//-----Student Login-----
$Password = sha1($NIC_Old); // "sha1" Encryption algorithm has used for the
Encryption
$User_Role = 'Student';

try {
    if (isset($_POST['Add'])) {

        //-----Start Handling Image File Uploads-----
        -----
        $allowedExts = array("gif", "jpeg", "jpg", "png");
        $temp = explode(".", $_FILES["Student_Image"]["name"]);
        $extension = end($temp);
        if (true) {
            if ($_FILES["Student_Image"]["error"] > 0) {
                echo "Return Code: " . $_FILES["Student_Image"]["error"] .
"<br>";
            } else {
                echo "Upload: " . $_FILES["Student_Image"]["name"] . "<br>";
                if (file_exists("upload/" .
$_FILES["Student_Image"]["name"])) {
                    echo $_FILES["Student_Image"]["name"] . " already exists.
";
                } else {
                    move_uploaded_file($_FILES["Student_Image"]["tmp_name"],
"upload/" . $_FILES["Student_Image"]["name"]);
                    echo "Stored in: " . "upload/" .
$_FILES["Student_Image"]["name"] . "<br>";

                    $Student_Image = $_FILES["Student_Image"]["name"];
                }
            }
        } else {
            echo "Invalid file";
        }

        //-----End Handling Image File Uploads-----
        -----

        //-----Create new student-----

        $sql = 'INSERT INTO
tbl_student(Registration_Number,Registration_Date,Title,Name_With_Initials,Fu
ll_Name,Date_of_Birth,NIC_Old,NIC_New,Gender,Mobile_Number,Resident_Number,Of
fice_Number,Email_Address,Address_Line1,Address_Line2,City,Postal_Code,Presen
t_Office,Supervising_Architect,Academic_Qualification,Counselor,Student_Image
,Admission_Form) '
        .
        'VALUES(:Registration_Number,:Registration_Date,:Title,:Name_With_Initials,:F
ull_Name,:Date_of_Birth,:NIC_Old,:NIC_New,:Gender,:Mobile_Number,:Resident_Nu
mber,:Office_Number,:Email_Address,:Address_Line1,:Address_Line2,:City,:Posta
l_Code,:Present Office,:Supervising Architect,:Academic_Qualification,:Counse
lor,:Student_Image,:Admission_Form)';
    }
}

```

- **Code Segment for Adding Course Fee Payment**

```

<?php

require_once '../control/config.php';

$Registration_Number = $_POST['Registration_Number'];
$Installment_Number = $_POST['Installment_Number'];
$Receipt_Number = $_POST['Receipt_Number'];
$Amount = $_POST['Amount'];
$Payment_Date = $_POST['Payment_Date'];

try {
    if (isset($_POST['Add'])) {

        //----- Validate the Amount to be paid
        $sqlV = "select Registration_number, sum(Amount) as Paid, (select
Amount from tbl_payment_definition where Fees_Type = 'Type_1') TotAmt from
tbl_payment_course where Registration_number = :Registration_Number";
        $stmtV = $conn->prepare($sqlV);
        $stmtV->execute(array(':Registration_Number' =>
$Registration_Number));
        $result = $stmtV->fetchAll();
        $row = $result[0];

        $Paid = $row['Paid'] + $Amount;
        $TotAmt = $row['TotAmt'];

        //-----Validate the Amount to be paid

        if ($Paid > $TotAmt) { //Payment Amount Validation
            $_SESSION['ERROR'][] = 'Error !!, Amount to be paid ' . $TotAmt.'
Total Amount to pay ' . $Paid;
        } else { // Insert the Course Payment Amount to table
            $sql = 'INSERT INTO
tbl_payment_course(Registration_Number,Installment_Number,Receipt_Number,Amou
nt,Payment_Date) '
            .
'VALUES (:Registration_Number,:Installment_Number,:Receipt_Number,:Amount,:Pay
ment_Date)';

            $stmt = $conn->prepare($sql);
            $stmt->execute(array('Registration_Number' =>
$Registration_Number,
            'Installment_Number' => $Installment_Number,
            'Receipt_Number' => $Receipt_Number,
            'Amount' => $Amount,
            'Payment_Date' => $Payment_Date));
            $conn = null;

            $_SESSION['SUCESSS'][] = 'New Course Payment Added Successfully:
Registration Number ' . $Registration_Number.' . Do you want to print the
payment Invoice? <a
href="CourseFeeInvoice.php?Receipt_Number=' . $Receipt_Number.'"
target="_blank" class="btn btn-info"> Print</a>';

        }
    }
}

```

- **Code Segment for Students Grades Calculation and Results Upload**

```

<?php
require_once '../..//control/config.php';

try {
    if (isset($_POST['Add'])) { //Adding Exam Results
        foreach ($_POST["Registration_Number"] as $rec => $value) {
            $Year = $_POST["Year"][$rec];
            $Registration_Number = $_POST["Registration_Number"][$rec];
            $Module_Code = $_POST["Module Code"][$rec];
            $Marks = $_POST["Marks"][$rec]; //Grading calculation method
            based on Students' marks
            if ($Marks >= 75) {
                $Grade = "A";
            } elseif ($Marks < 74 && $Marks >= 60) {
                $Grade = "B";
            } elseif ($Marks < 59 && $Marks >= 40) {
                $Grade = "C";
            } elseif ($Marks < 39 && $Marks > 0) {
                $Grade = "F";
            } else {
                $Grade = "NA";
            }

            $Comments = $_POST["Comments"][$rec];
            $Effective_Date = $_POST["Effective Date"][$rec];
            $Approved_By = $_POST["Approved_By"][$rec];
            $sql = 'INSERT INTO
tbl_exam_results(Year,Registration_Number,Module_Code,Marks,Grade
,Comments,Effective_Date,Approved_By) '
            .
            'VALUES(:Year,:Registration_Number,:Module_Code,:Marks,:Grade,:Co
mments,:Effective_Date,:Approved_By)';

            $stmt = $conn->prepare($sql);
            $stmt->execute(array('Registration_Number' =>
$Registration_Number,
'Year' => $Year,
'Module_Code' => $Module_Code,
'Marks' => $Marks,
'Grade' => $Grade,
'Effective_Date' => $Effective_Date,
'Comments' => $Comments,
'Approved_By' => $Approved_By));
        }
        $conn = null;

        $_SESSION['SUCESSS'][] = 'Exam Results Successfully Added for:
Registration_Number ' . $Registration_Number;
    }
}

```

- Base CSS Code for PDF Invoices

```
.invoice-box{
  max-width:800px;
  margin:auto;
  padding:30px;
  border:1px solid #eee;
  box-shadow:0 0 10px rgba(0, 0, 0, .15);
  font-size:16px;
  line-height:24px;
  font-family:'Helvetica Neue', 'Helvetica', Helvetica, Arial, sans-
  serif;
  color:#555;
}

.invoice-box table{
  width:100%;
  line-height:inherit;
  text-align:left;
}

.invoice-box table td{
  padding:5px;
  vertical-align:top;
}

.invoice-box table tr td:nth-child(2){
  text-align:right;
}

.invoice-box table tr.top table td{
  padding-bottom:20px;
}

.invoice-box table tr.top table td.title{
  font-size:45px;
  line-height:45px;
  color:#333;
}

.invoice-box table tr.information table td{
  padding-bottom:40px;
}

.invoice-box table tr.heading td{
  background:#eee;
  border-bottom:1px solid #ddd;
  font-weight:bold;
}

.invoice-box table tr.details td{
  padding-bottom:20px;
}

.invoice-box table tr.item td{
  border-bottom:1px solid #eee;
}
```



- **Code Segment for Payment Summary Reports**

```

<?php
// get the HTML
ob_start();
require_once '../../control/config.php';
?>

<link href="ReportStyle.css" rel="stylesheet" type="text/css"/>
<page>
<br> //Show SLIA Logo
-----
<h1>Course Fee Payment Summary Report</h1><br>
<h5>Report Duration: <?php echo $_POST['Payment_FDate'];?> - <?php echo
$_POST['Payment_TDate'] .' User : ' . $_SESSION['Name_With_Initials'];?>
</h5><br>

<?php

$Payment_FDate = $_POST['Payment_FDate']; //Payment From Date
$Payment_TDate = $_POST['Payment_TDate']; //Payment To Date

$sql = "select A.Registration_Number, B.Name_With_Initials,
A.Installment_Number, A.Receipt_Number, A.Amount, A.Payment_Date
from tbl_payment_course A, tbl_student B
where A.Registration_Number = B.Registration_Number
and A.Payment_Date between :Payment_FDate and
:Payment_TDate";

$stmt = $conn->prepare($sql);
$stmt->execute(array(
    ':Payment_FDate' => $Payment_FDate,
    ':Payment_TDate' => $Payment_TDate
));

echo '<table class="table">';
echo '<thead>';
echo '<tr>';
    echo '<th>Registration Number</th>';
    echo '<th>Name With Initials</th>';
    echo '<th>Installment Number</th>';
    echo '<th>Receipt Number</th>';
    echo '<th>Amount</th>';
    echo '<th>Payment Date</th>';
echo '</tr>';
echo '</thead>';

echo '<tbody>';
while ($row = $stmt->fetch(PDO::FETCH_NUM))
{
    echo '<tr>';
    echo '<td>'.$row[0]. '</td>';
    echo '<td>'.$row[1]. '</td>';
    echo '<td>'.$row[2]. '</td>';
    echo '<td>'.$row[3]. '</td>';
    echo '<td>'.$row[4]. '</td>';
    echo '<td>'.$row[5]. '</td>';
    echo '</tr>';
}
}

```

# Appendix G - Client Certificate

Sri Lanka  
Institute of  
Architects



- 120/7, Vidya Mawatha,  
Colombo 7,
- Sri Lanka.
- Tel : (+94-11) 2678255, 2697109, 2697170
- Fax : (+94-11) 2682757
- E-mail : managerbae@slia.info,  
bael@slia.info
- Web : www.slia.lk



06<sup>th</sup> November 2017

Projects Examination Board  
University of Colombo School of Computing  
221/2A, Dharmapala Mawatha  
Colombo 7.

Dear Sir/Madam,

## LETTER OF CERTIFICATION

Re. the "Students Management System for SLIA" developed by Mr. P.M.N. Sampathpriya (R020168)

I wish to inform the members of the Board that the Students Management System developed by Mr. Sampathpriya was shown to the Board of Architectural Education (BAE) of the Sri Lanka Institute of Architects (SLIA) and we are convinced that once the system implementation is completed with the areas we highlighted it will serve the management purposes of the BAE. The gathering of information and analysis of the present system was done in a meticulous manner and we are quite impressed by the systematic approach followed by Mr. Sampathpriya in developing the system. His determination to perfect the system is praiseworthy.

We will continue to use the regular system that we have been using so far for another year in parallel with the web based system developed by Mr. Sampathpriya and will seek his assistance to fix any operational errors that might encounter during the parallel run. We are thankful for him for selecting this project as his final year project for the BIT Degree and we are also happy to note the very high standards the University demands from its students which augers well for the future of the country.

This letter was issued as requested by Mr. Sampathpriya.

Thanking you  
Yours faithfully

  
Archt. Veranjan Kurukulasuriya  
Senior Vice President

### OFFICE BEARERS FOR SESSIONS 2017/2018 & 2018/2019

Archt. D. H. Wijewardene, FIA (SL) - PRESIDENT

Archt. Harsha Fernando, FIA (SL) - IMMEDIATE PAST PRESIDENT

Archt. Veranjan Kurukulasuriya, FIA (SL) - Senior Vice President  
Archt. Rohana Bandara Herath, FIA (SL) - Hony. Secretary  
Archt. Michele Fonseka, FIA (SL) - Hony. Asst. Secretary

Archt. Russell Dandenya, FIA (SL) - Vice President  
Archt. Sallinda Rathugama, AIA (SL) - Hony. Treasurer  
Archt. Migara Alwis, FIA (SL) - Hony. Asst. Treasurer

# GLOSSARY

- Android** - Android is an operating system developed by Google Company mainly focusing on mobile devices.
- Bootstrap Framework** - Bootstrap is a free and open-source front-end framework for designing websites and web based applications.
- iOS** - iOS is an operating system created and developed by Apple Inc. exclusively by targeting Apple branded mobile devices.
- jQuery** - This is a cross-platform JavaScript library. It is designed to simplify the client-side scripting of HTML.
- Encryption Function SHA-1** - Stands for Secure Hash Algorithm-1, which is a cryptographic hash function designed by the United States National Security Agency.
- StarUML** - Star UML is open-source software which is used to create UML diagrams.

# INDEX

## A

ABSTRACT iii  
Architect iii  
Act iii  
Authority iii  
Architecture 1,21,40,47  
Amendment 1

## B

Bandwidth 13  
Board iii  
Bootstrap iii

## C

Centralized iii  
Cumbersome iii  
Client iii  
Chartered 1  
Consistency 7

## D

Degree ii  
Desktop 23  
Database 28, 36  
Documentation 54, 58

## E

Education iii  
Engineering 7  
Encryption 34

## F

Functionalities 10, 13  
,14,29,44,48,69,89  
Frameworks 23, 38

## G

Greece 1

## H

History 1

## I

Institute iii  
Incorporated iii  
Interviews 7  
Interfaces 7

## K

Knowledge ii  
Kindergartens 12

## L

Literary 1  
Law 1  
Legislation 1  
Licentiate 1

## M

Modules iii, 47  
Methodologies 18, 20, 48

## N

Notification iii  
Non-functional 17

## O

Organizations ii  
Observation 8  
Open-source 10, 33, 96

## P

Password 59  
Professional iii  
Parliament iii  
Pivotal 1  
Privileges 46, 62

## Q

Questionnaires 8, 88  
Qualifications 47, 78

## R

Rational 18, 19  
Reference ii, 50

## R

Reports iii,74,75,76,77  
Requirement 7,14  
,21,32,39,40,48  
Reusable 38

## S

Spectrum iii, 1,  
Streamlining iii, 1,  
Statute 1  
Software 7

## T

Title ii  
Templates iii  
Troubleshooting 23  
Transcripts 47

## U

University ii,47  
User-friendly 11,18  
Unanimously 23  
Unauthorized 62

## V

Validate 42,44  
Verify 42,44  
Verified 46  
Visualization 38,78  
Verification 61

## W

Windows 23, 53  
Web-based 52

## X

XAMPP 32,33,52,53