



**WEB BASED RESULTS MANAGEMENT  
SYSTEM FOR FACULTY OF  
MANAGEMENT STUDIES & COMMERCE,  
UNIVERSITY OF JAFFNA**

S.M.M. JEEWANTHA

Registration Number: R160017

Index Number: 1600176

Name of the supervisor:

Dr. E.Y.A. Charles

**October 2018**



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.

Signature of Candidate: .....

Date: 03/10/2018

Name of Candidate: Mr. S.M.M. Jeewantha

Countersigned by:

Signature of Supervisor/Advisor: .....

Date: 03. Oct 2018.

Name of Supervisor/Advisor: Dr. E.Y.A. Charles

Dr. E. Y. A. Charles  
B.Sc (Jaffna). Ph.D (Wales)  
Senior Lecturer in Computer Science &  
Head/ Computer Centre  
University of Jaffna, Sri Lanka.

# **Abstract**

The Faculty of Management Studies and Commerce of University of Jaffna was established in May 1999. The faculty is currently divided into five departments and offers two main undergraduate courses. Manual methods and spreadsheet programs are being used to manage and publish students results after every semester exam. Due to the current system they face delay in publishing, managing problems and students do not have any place to track their progress.

By the proposed web based system students can access to their past results, GPA marks and they may get some guidance to achieve their academic carrier. Faculty staff and examination branch staff can use system to publish and manage student results, generate various reports and authorized persons can maintain and view student and lecturer profiles.

Rational Unified Process was selected as the process model among all other software development methodologies by considering its advantages over other process models. Object oriented analysis and design techniques were used for the project and UML was used for modelling the system. The system was developed using MVC [17] architecture with Laravel [16] framework version 5.5 and used responsive design. Apache 2.4 was used as webserver. Object oriented PHP 7.1 for server side scripting, JavaScript for client side scripting, MySQL 5.7 for database and HTML5, CSS, Bootstrap [13] for the interface design were used.

After implementing the system results publishing process take less time and managing results will be an easy task. Student and lecturer profiles data can be used for various purposes without consuming additional time. Apart from that, students can monitor and plan their academic carrier activities effectively.

# Acknowledgement

The success and final outcome of this project required a lot of guidance and assistance from many people and I am extremely privileged to have got this all along the completion of my project. All that I have done is only due to such supervision and assistance and I would not forget to thank them.

First of all, I would like to express my profound gratitude to BIT Coordinator of the UCSC and to the Project Examination Board for giving me this valuable opportunity to follow a world recognized degree program.

My sincere gratitude also goes to Dr. E.Y.A. Charles, senior lecturer of Department of Computer Science, University of Jaffna who was also my technical supervisor. His valuable guidance and dedication helped me to implement my project to a reality. I am gratefully obliged to Dr. E.Y.A. Charles for his cordial support and the guidance.

I respect and thank Prof.T.Velnampy, The Dean, Faculty of Management Studies & Commerce, University of Jaffna, and Prof.R.Vigneswaran, The Vice Chancellor, University of Jaffna, for providing me an opportunity to do the project work in Faculty of Management Studies & Commerce, University of Jaffna and giving me all support and guidance which made me complete the project duly. And I am extremely thankful to the staff in examination branch for providing such a nice support and guidance.

I also extend my heartfelt thanks to my parents for their support without which my success of the project would have been a distant reality. I am obliged to my family members for the constant support, encouragement and the guidance given, which will carry me a long way in the journey of life, in which I am about to embark.

I would not forget to remember Mr.B.Prahalathan, The Head of the Department, Department of Commerce, Faculty of Management Studies & Commerce, University of Jaffna, and other staff members of Department of Commerce for their encouragement and more over for their timely support and guidance till the completion of project work.

Finally, I would like to thank everyone specially my friends for encouraging and helping me to complete this project successfully.

# Table of Contents

<b>Declaration.....</b>	ii
<b>Abstract.....</b>	iii
<b>Acknowledgement .....</b>	iv
<b>List of Figures.....</b>	viii
<b>List of Tables .....</b>	x
<b>List of Acronyms .....</b>	xi
<b>Chapter 1 - Introduction .....</b>	1
1.1    About University of Jaffna.....	1
1.2    About Faculty of Management Studies & Commerce .....	1
1.3    Web Based Results Management System.....	2
1.4    Motivation of the Project .....	3
1.5    Scope of the Project .....	3
1.6    Objectives of the Project .....	4
1.7    Structure of the Dissertation.....	4
<b>Chapter 2 - Analysis.....</b>	6
2.1    Introduction.....	6
2.2    Requirement Gathering Techniques.....	6
2.3    Analysing the Current Manual System .....	7
2.3.1    Student Registration .....	9
2.3.2    Evaluation Process .....	9
2.3.3 Completing a Course.....	9
2.4    Drawbacks of the Current System.....	10
2.5    Functional Requirements of the System .....	10
2.6    Non-Functional Requirements of the System .....	11
2.7    Similar Systems and Literature Review .....	12
<b>Chapter 3 - Design.....</b>	14
3.1 Web-based Solution .....	14
3.2    Overview of Process Models .....	14
3.2.1    Process Model for the System.....	15
3.3    Object Oriented Analysis and Design .....	16
3.3.1    High Level Use Case Diagram for the Results Management System .....	17
3.3.2    Class Diagram for the Results Management System .....	18
3.4    Database Design.....	19

3.5 User Interface Design.....	20
3.5.1 Key Design Considerations Used for the System .....	20
3.5.2 Login Interface .....	23
3.5.3 Configuration Module Interfaces .....	24
3.5.4 Dashboard Interfaces.....	25
<b>Chapter 4 - Implementation.....</b>	<b>26</b>
4.1 Introduction .....	26
4.2 Implementation Environment.....	26
4.3 System Development Tools and Technologies .....	27
4.3.1 Visual Studio Code .....	27
4.3.2 PHP .....	27
4.3.3 Laravel .....	27
4.3.4 JavaScript .....	28
4.3.5 JQuery .....	28
4.3.6 Ajax.....	28
4.3.7 CSS.....	28
4.3.8 Bootstrap .....	28
4.3.9 MySQL.....	29
4.3.10 Apache HTTP Server .....	29
4.3.11 Adobe Photoshop CC 2018.....	29
4.4 Major Code Segments.....	30
4.4.1 Middleware Attachment.....	30
4.4.2 Validations .....	30
4.4.3 GPA Calculation Trait.....	31
4.4.4 Student Model .....	34
4.4.5 Delete Function via Ajax .....	36
4.4.6 Send Mail Job.....	36
4.5 Re-used Components .....	38
4.5.1 Toastr - JavaScript Notification .....	38
4.5.2 Bootstrap select .....	38
4.5.3 Laravel-DOM pdf .....	38
4.5.4 Laravel Excel .....	38
4.5.5 Laravel-chartjs .....	39
4.5.6 SB Admin – Dashboard Template .....	39
4.6 Module Structure.....	39
<b>Chapter 5 - Evaluation .....</b>	<b>42</b>

5.1 Introduction .....	42
5.2 System Test Plan and Test Cases .....	42
5.3 User Acceptance Testing .....	45
<b>Chapter 6 - Conclusion .....</b>	<b>50</b>
6.1 Introduction .....	50
6.2 Critical Evaluation of the Project .....	50
6.3 Future Improvements .....	52
6.4 Lessons Learnt .....	52
<b>References .....</b>	<b>54</b>
<b>Appendix A – System Manual .....</b>	<b>57</b>
<b>Appendix B – Design Documentation .....</b>	<b>60</b>
<b>Appendix C – User Manual .....</b>	<b>65</b>
<b>Appendix D – Management Reports .....</b>	<b>84</b>
<b>Appendix E – Test Results .....</b>	<b>90</b>
<b>Appendix F – Code Listing .....</b>	<b>101</b>
<b>Appendix G – Client Certificate .....</b>	<b>106</b>
<b>Glossary .....</b>	<b>107</b>
<b>Index .....</b>	<b>109</b>

# List of Figures

Figure 2.1 Use Case Diagram for Manual System.....	8
Figure 2.2 Admin Dashboard – “OpenSIS Pro” .....	12
Figure 2.3 Admin dashboard - "TotalSchool".....	13
Figure 3.1 Rational Unified Process .....	16
Figure 3.2 High Level Use Case Diagram .....	17
Figure 3.3 Class Diagram.....	18
Figure 3.4 Database Design .....	19
Figure 3.5 Notifications .....	20
Figure 3.6 Modal Pop-up Windows .....	21
Figure 3.7 Meaningful Error Messages.....	21
Figure 3.8 Password Validation .....	22
Figure 3.9 Dynamic Components .....	22
Figure 3.10 Help Blocks .....	22
Figure 3.11 Breadcrumb .....	23
Figure 3.12 Login Interface .....	23
Figure 3.13 Dean Profile Creation - Configuration Interface .....	24
Figure 3.14 Dashboard of Dean .....	25
Figure 4.1 Formula for GPA calculation.....	31
Figure 5.1 User evaluation questionnaire - Lecturer.....	46
Figure 5.2 User evaluation questionnaire - Student .....	47
Figure 5.3 User evaluation questionnaire - Examination Branch .....	48
Figure 5.4 Summary of user friendliness .....	49
Figure 5.5 Summary of overall performance .....	49
Figure 6.1 Summary of user acceptance test.....	51
Figure A.1 File structure of the system.....	58
Figure B.1 Use case diagram for result publishing .....	60
Figure B.2 Use case diagram for managing student details .....	62
Figure B.3 Use case diagram for system configuration .....	64
Figure C.1 Dean’s profile creation.....	65
Figure C.2 AR profile creation .....	66
Figure C.3 AR profile creation - Default password received from the system .....	66
Figure C.4 Examination branch profile creation.....	67
Figure C.5 Course creation .....	67
Figure C.6 Department creation.....	68
Figure C.7 Specialized areas creation .....	68
Figure C.8 Subjects creation .....	69
Figure C.9 Grading system creation.....	69
Figure C.10 Academic year creation.....	70
Figure C.11 Log in interface of the system.....	70
Figure C.12 Reset log in password .....	71
Figure C.13 Too many login attempts error.....	71
Figure C.14 Change password .....	71
Figure C.15 Navigation pane of the system.....	72
Figure C.16 Add new student to the system .....	73
Figure C.17 Add new lecturer to the system.....	74
Figure C.18 Department details .....	75
Figure C.19 Add new course to the system.....	76

Figure C.20 Subject details .....	77
Figure C.21 Add new results.....	77
Figure C.22 Add new results - Enter new grades.....	78
Figure C.23 Notification - Results added message .....	78
Figure C.24 Edit student results .....	78
Figure C.25 Add repeating subject results .....	79
Figure C.26 Publish results .....	79
Figure C.27Search options - Student Results.....	80
Figure C.28 Full results view - Part one .....	80
Figure C.29 Full results view - Part two .....	80
Figure C.30 Dashboard - Dean .....	81
Figure C.31 Dashboard - Student.....	81
Figure C.32 Settings page .....	82
Figure C.33 Reports - Selection of columns .....	83
Figure D.1 Reports - Students in specific academic year .....	84
Figure D.2 Reports - Students in specific course .....	84
Figure D.3 Reports - Students in specific department .....	85
Figure D.4 Reports - Students in specific specialized area.....	85
Figure D.5 Reports - Full details of a student .....	85
Figure D.6 Reports - Assigned subjects of a student .....	86
Figure D.7 Reports - All the lecturers in the faculty.....	86
Figure D.8 Reports - Lecturers in specific department .....	86
Figure D.9 reports - Assigned subjects of a lecturer.....	87
Figure D.10 Reports - Full results of a student .....	87
Figure D.11Reports - Subject results .....	88
Figure D.12 Reports - Semester Results .....	88
Figure D.13 Reports - Course details.....	89
Figure D.14 Reports - Grading system .....	89

# List of Tables

Table 4.1 Implementation Environment .....	26
Table 5.1 High level test plan .....	45
Table A.1 Hardware requirements .....	57
Table A.2 Software requirements .....	57
Table B.1 Use case description - Publishing results .....	61
Table B.2 Use case description - Access to the publishing results .....	61
Table B.3 Use case description - Maintain student details .....	63
Table B.4 Use case description - Assign specialization .....	63
Table B.5 Use case description - System configuration .....	64
Table E.1 Test Cases for Configuration Module .....	90
Table E.2 Test Cases for Settings Module.....	91
Table E.3 Test Cases for Notification Module .....	92
Table E.4 Test Cases for Courses Module.....	92
Table E.5 Test Cases for Departments Module .....	93
Table E.6 Test Cases for Authentication Module .....	94
Table E.7 Test Cases for Profiles Module .....	95
Table E.8 Test Cases for Grades Module .....	96
Table E.9 Test Cases for Lecturer Module .....	97
Table E.10 Test Cases for Student Module .....	99
Table E.11 Test Cases for Subjects Module .....	99
Table E.12 Test Cases for Reports Module .....	100

# **List of Acronyms**

AR – Assistant Register

CSS - Cascading Style Sheets

ER Diagram – Entity Relationship Diagram

FMSC – Faculty of Management Studies & Commerce

GPA – Grade Point Average

HTML – Hypertext Markup Language

HTTP – Hyper Text Transfer Protocol

IDE – Integrated Development Environment

PHP - Hypertext Preprocessor

RUP – Rational Unified Process

# Chapter 1 - Introduction

## 1.1 About University of Jaffna

The University of Jaffna is a public university in the city of Jaffna in Sri Lanka with the motto of “**Discernment is wisdom**”. Established in 1974 as the sixth campus of the University of Sri Lanka, it became an independent, autonomous university in January 1<sup>st</sup> 1979. The chancellor and vice-chancellor are professors S. Pathmanathan and R. Vigneswaran respectively.

University of Jaffna has two campuses - the main campus in Thirunelveli in Jaffna and a second campus in Vavuniya. It has ten faculties (Agriculture, Applied Science, Arts, Business Studies, Engineering, Graduate Studies, Management Studies & Commerce, Medicine, Science and Technology) and thirteen other academic units/centres. The university offers undergraduate and postgraduate courses that award various degrees.

The university had 7,972 students and 1,342 employees in 2016. It is the seventh largest university in Sri Lanka in student numbers [1, 2].

## 1.2 About Faculty of Management Studies & Commerce

The Faculty of Management Studies and Commerce was established in May 1999 from parts of the Faculty of Arts. Professor M. Nadarajasingham, Head of the Department of Management Studies, was appointed the first Dean of the Faculty of Management Studies and Commerce.

The faculty offers undergraduate courses which lead to Bachelor of Business Administration (B.B.A.) and Bachelor of Commerce (B.Com.) degrees.

Currently Bachelor of Business Administration (B.B.A.) provide specialized in accounting, finance management, human resource management and marketing management, and Bachelor of Commerce (B.Com.) provide specialized in accounting & finance as well as business technology management areas.

The faculty is currently divided into five departments,

- Department of Accounting
- Department of Financial Management
- Department of Human Resource Management
- Department of Marketing
- Department of Commerce

The faculty's current dean and assistant registrar are T. Velnampy and K. Vijitha respectively. The faculty had 1,406 undergraduate students and 48 academic staff (40 permanents, 8 temporary) in 2016.

### **Vision**

“To provide best possible human resources to society in the field of Management and Commerce.”

### **Mission**

“Our graduates will be amongst those who exercise effective Leadership in tomorrow’s business world due to their ability to sense possibilities to think among different perspectives and the smart way of doing things with a sense of responsibility and innovation and create strategic thinking in the Market Place” [1, 2, 3].

## **1.3 Web Based Results Management System**

Currently, traditional methods such as physical documents, spreadsheet programs are being used by FMSC to maintain student records and their results. When speeding up results publishing process and provide easy access of results to students that modern technology support is important. So, an automated web based results management system was needed to handle the results publishing and managing process smoothly.

## 1.4 Motivation of the Project

Students face difficulty to track their progress in academic carrier because the results publishing process take long time to publish students results due to the manual system. Even some students do not know which subjects to be repeated, what is the current state of GPA, did they sit before that subject etc. Then the students going to clarify their issues entering dean's office or department. At that time, they also having trouble in finding the solutions because they must use manual system. It will take more time and there is a probability to occur errors. The faculty should maintain big files according to academic years and according to each course unit which is require big paper works and time.

And, students need a place to track GPA to achieve their academic carrier path. Therefore, the FMSC needed web based system which can access by students and university staff.

Thus, the new system was developed to support the FMSC to handle their results publishing operations successfully by overcoming the above problems, which occurred due to using the file based system.

## 1.5 Scope of the Project

When deciding to develop a system, the scope is a most important fact which we need to consider about. The scope of this project will be as follows.

- System has provided to store student details, lecturer details and department details of the faculty.
- System has facilitated to store student final results by examination department of the university which is accessed by each student as well as authorized persons.
- When results have been published students will get notifications and they can access to updated GPA marks.

## 1.6 Objectives of the Project

An automated computer based system was developed to perform their results publishing and managing activities accurately and efficiently to overcome their current drawbacks was the main objective. Apart from the above, other objectives are listed below:

- Provide a login module for faculty staff as well as students to see their results.
- Provide an automated place to track GPA for each student.
- Improve the coordination between faculty staff and students.
- System has facilitated a complete student profile to authorized people.
- User management will secure the system data very well.
- System can produce complete result sheet for each student.
- Report generating module provide variety of reports.
- Dashboard is provided to administrators for decision making purposes.

## 1.7 Structure of the Dissertation

Six main chapters with the Introduction Chapter are comprised in this report. Six main chapters are briefly described as follows:

### Chapter 2: Analysis

Requirement gathering techniques such as interviews and observations are described in this chapter. How the current system works and what the requirements are for the project (functional requirements and non-functional requirements) are also identified here. UML diagrams such as use case diagrams are drawn and provided to identify un-clear requirements and obtain better ideas about the system.

## **Chapter 3: Design**

According to the analysed requirements, the database design, user interface design and other design methods are carried out. Sample interfaces and database design diagram are included in this chapter.

## **Chapter 4: Implementation**

The system specification conversion into practical world is described in this chapter using different web supported languages. Implementation environment, used technologies, reused modules are discussed in this chapter.

## **Chapter 5: Evaluation**

How the implement system should be tested is described in this chapter; such as test cases planning, applying different testing methodologies to test for the accuracy of the system. User acceptance testing is also carried out at this stage.

## **Chapter 6: Conclusion**

Lessons learnt by implementing this system with a brief description of the company and future improvements of the system are discussed in this chapter.

## **References**

All the URLs references and necessary quotations which helped to write this report are contained in this section.

## **Appendix**

Consists of seven topics; each topic describes the system in detail. This section has been written in detail for the interested parties to learn about the system.

# **Chapter 2 - Analysis**

## **2.1 Introduction**

“Requirements analysis, also called requirements engineering, is the process of determining user expectations for a new or modified product. These features, called requirements, must be quantifiable, relevant and detailed” [4].

Requirement gathering techniques used to collect requirements, how the current system process is going on, draw backs of the present system, functional and non-functional requirements of the proposed system and finally the similar system reviews are contained in this chapter.

## **2.2 Requirement Gathering Techniques**

Requirements gathering is very important and critical process which should complete before the analysis phase. Inadequate information would cause waste of time for modifications and lose the opportunity and would cause additional unnecessary overhead.

There are several requirements gathering techniques which can be used to collect the clear and accurate information. In this project requirements were gathered by using following techniques.

- Interviews

Several interviews were conducted with the dean FMSC, the head of the department of commerce and some lectures of the faculty and several officers in examination branch as the main requirement gathering technique.

- Observations

Through observation it could be understood about the overall process of results publishing and its intermediary works.

- Documentations

Number of documents reviewed to understand the manual system of department of commerce. For example, format of academic transcript.

## 2.3 Analysing the Current Manual System

The Faculty of Management Studies and Commerce is organized under the dean. The dean's office coordinates all academic and administrative activities within the Faculty. There is a head appointed for each academic department that comes under the faculty. The heads of the department report to the dean. To assist the dean in the administrative work an assistant registrar(AR) is appointed. He or she is responsible for assisting dean in all matters regarding non- academic, registration of students in the degree programme, examination work, and secretarial work of the Faculty Board, and other general administration work. Final results publishing, transcript generation and GPA calculation is done by the examination department of university [3].

Figure 2.1 shows the use case diagram of the current manual system.

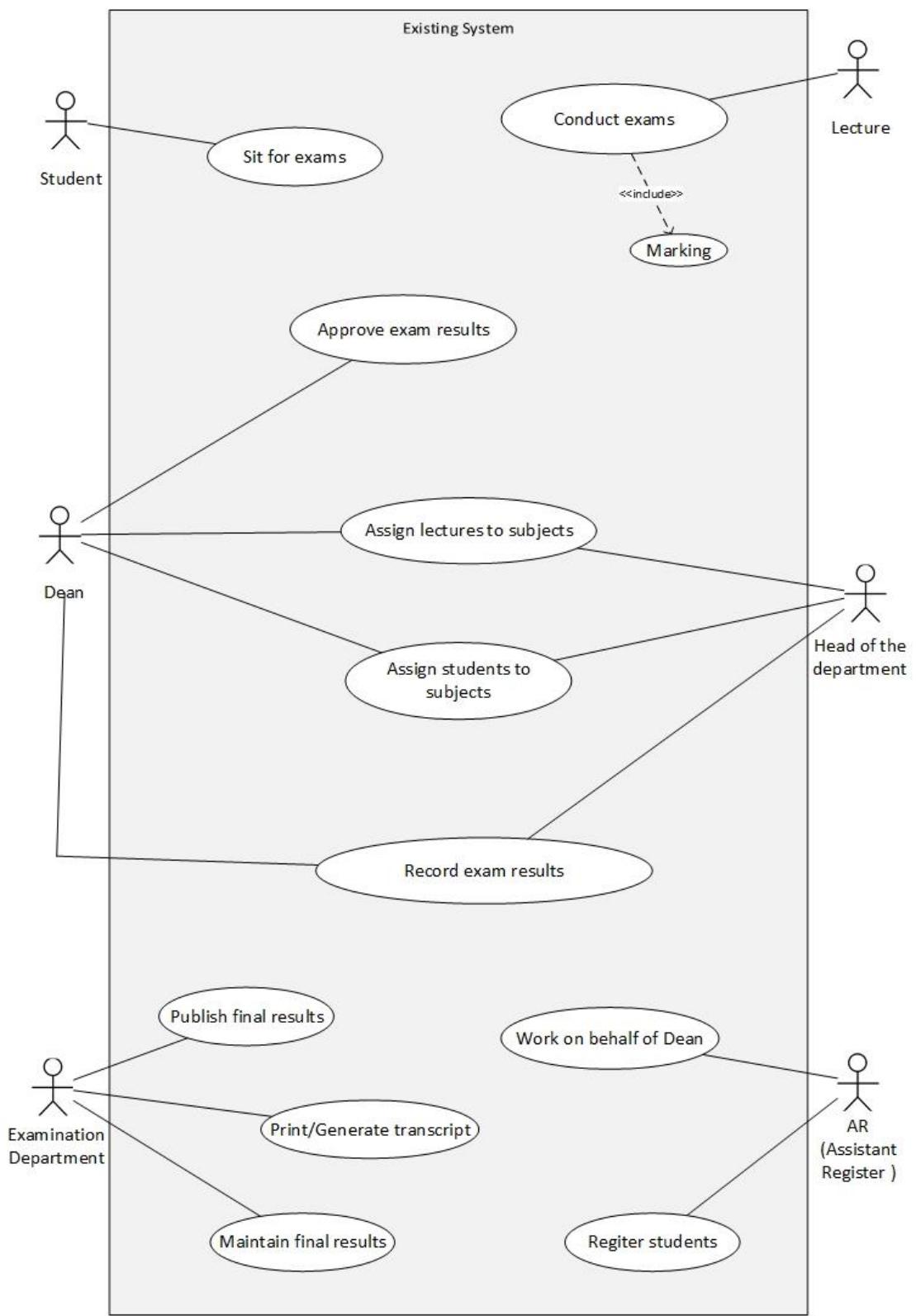


Figure 2.1 Use Case Diagram for Manual System

### 2.3.1 Student Registration

As the first step students are registered by the faculty. This process is highly based on manual system. The faculty is maintaining student record for each student. This may include student's personal details, academic details, family details, special details in some cases. If someone needed to access this information he or she should have access physically the files maintained by the faculty.

### 2.3.2 Evaluation Process

Students who obtain a grade of less than 'B' or did not participate may repeat that course by registering that course in the subsequent semester [3]. Minimum pass level is 'C' grade.

### 2.3.3 Completing a Course

If a candidate absents for end semester examination of a course, the following symbol will be indicated appropriately.

- Absent due to medical reason, which has been approved, will be given symbol of "MC" (medical).
- Absent due to valid reasons, which has been approved will be given symbol to "DFR" (Deferred).
- Absent without valid reasons will be given symbol of "AB" (Absent).

The above symbols (MC, DFR, AB) will not be considered for credit calculation [3].

After the examination board meetings, final marks will be handover to the examination branch for publish final results (Handover as a hard copy is the official way). When results published by the examination branch students can get printed result sheet for their references. GPA calculation can be done by students based on that printed results sheet. The faculty is not providing any automated tools to calculate and track students' progress.

## 2.4 Drawbacks of the Current System

The following major drawbacks that slow down the results publishing process and the related sub process while gathering requirements were reported by the Dean of the FMSC and the responsible members of the administration:

- Students and other authorized parties does not have instant results viewer, if someone needed they should search in physical file based system.
- There is no proper place to track students' progress.
- Faculty and its departments separately maintain student marks which may increase data duplication.
- The faculty does not have complete student profile.
- Due to the current system it may increase paper usage.
- Due to the manual system results publishing process take more time.

## 2.5 Functional Requirements of the System

“In Software engineering and systems engineering, a functional requirement defines a function of a system or its component. A function is described as a set of inputs, the behaviour, and outputs” [7].

When it comes to the functional requirements it is essentially specifies something the system should do. Following are a list of functional requirements that are identified during the analysis stage:

- Login module according to the user type with different privileges and functionalities.
- Master file module.  
Users can insert, delete, update, view and search student details and lecturer details. But the functionalities can different based on user type.
- Notification Module.  
Students are get notified when new result has been published and notify to other users when they need attention.

- Report generating Module is only for authorized users according to their needs.
- Results management module

This is the main module of this system. It takes care of storing student results periodically which can be accessed by authorized persons.

- Settings module provide main configurations functions for authorised users.
- Course units' module

This provides managing course units according to different degree programmes to relevant users.

- Department module provide dynamic creation of departments to the faculty which can be done through authorised users.
- GPA calculation module provide algorithm for generating GPA of each student and notify to students and relevant authorities when they need attention.
- Lecturer - course unit enrolment module provide facility to make relationship between lectures and course units.
- Student - course unit enrolment module provide facility to enrol students to course units.

## 2.6 Non-Functional Requirements of the System

“In systems engineering and requirements engineering, a non-functional requirement (NFR) is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviours” [8].

Success of this system is not only dependent on the functional requirements; it is also based on the non-functional requirements. It acts as a main role for the success of the system. Following are the major non-functional requirements which were identified at this stage:

- Accuracy

Accuracy is an important non-functional requirement which we need to consider. If we do not consider about the accuracy, database problems can be occurred. By applying validation techniques to the user input data accuracy has been achieved in this system.

- User-friendliness

Overall system should be simple as much as possible with more user-friendly environment. This has been achieved by using various techniques such as easy menu navigation, searching options, attractive interfaces and use of matching colours.

- User accounts and access privileges should be maintained accurately to prevent unauthorized access.

## 2.7 Similar Systems and Literature Review

When we considering results management system for university level there is no exact system which can be satisfy the requirements. There are lots of school management systems out there. For example, “OpenSIS Pro” [5], “TotalSchool” [6]. Following figure 2.2 shows the admin dashboard of “OpenSIS Pro” software and figure 2.3 shows the admin dashboard of “TotalSchool” software.

But the requirements and process are different than university results management and grading system.

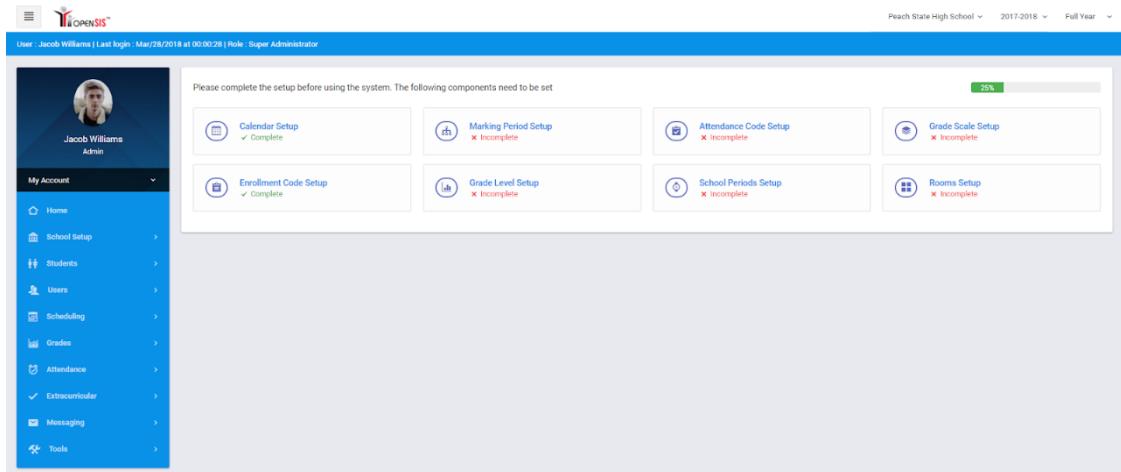


Figure 2.2 Admin Dashboard – “OpenSIS Pro”

PRO SCHOOLS

Admin Panel >

**Overview**

Announcement >

Classes >

Subjects >

Students >

Study Material >

Finance >

Invoices >

Behaviour >

Library >

Reports >

Settings

**Students**: 3

**Teachers**: 1

**Classes**: 2

**Online**: 1

**Finance**

Total Revenue	Total Expenses	Profit
\$36	\$0	\$36

**Classes This Week**

News & Updates

- Testing
- Headmaster Warning
- Rule Update
- School Operations

Your Assignments

- Test Assignment Due on 01/03/2017
- Exam One Due on 04/03/2017
- Yolo Due on 09/03/2017

Figure 2.3 Admin dashboard - "TotalSchool"

# **Chapter 3 - Design**

“System design is the process of defining the elements of a system such as the architecture, modules and components, the different interfaces of those components and the data that goes through that system. It is meant to satisfy specific needs and requirements of a business or organization through the engineering of a coherent and well-running system” [9].

There are several process models being used in the current software industry. In this chapter mainly discuss selected process model of object oriented design techniques throughout development, diagrams which describe the system and some main interfaces which deal with the system.

## **3.1 Web-based Solution**

Web-based solution is software we use over the internet with a web browser. We don't have to install anything, download any software or worry about upgrades. Responsive web based system has been proposed instead of standalone system for results management system because of the following reasons:

- We can access to the system anytime in anywhere.
- Web systems are platform independent.
- No need to install special software in client side, they just need a web browser only.
- Maintaining web based system is easy than standalone system.
- Implementing web based solutions may reduce cost of hardware.

## **3.2 Overview of Process Models**

Process model is a description of the sequence of activities carried out in software development project, and the relative order of these activities. Selecting the right process model is very important to success of project with desired requirements. This is heavily

depending on the kind of project and its requirements. There are various process models such as Waterfall, Agile, Incremental etc.

### 3.2.1 Process Model for the System

Rational Unified Process(RUP) was selected as the process model among all other software development methodologies by considering its advantages over other process models. Some important advantages are:

- RUP allows for the adaptive capability to deal with changing requirements throughout the development life cycle.
- It is an object oriented and web enabled development methodology.
- Throughout the development it ensures the software quality as well as the standard.
- Develop software iteratively.
- Risk management can be done easily.
- Use Unified Modelling Language(UML) to address the static view as well as the dynamic view of the system.

As the figure 3.1 shows it divides the development process into four distinct phases.

- **Inception Phase**

The goal of the inception phase is to establish a business case for the System. You should identify all external entities (people and systems) that will interact with the system and define these interactions [10].

- **Elaboration Phase**

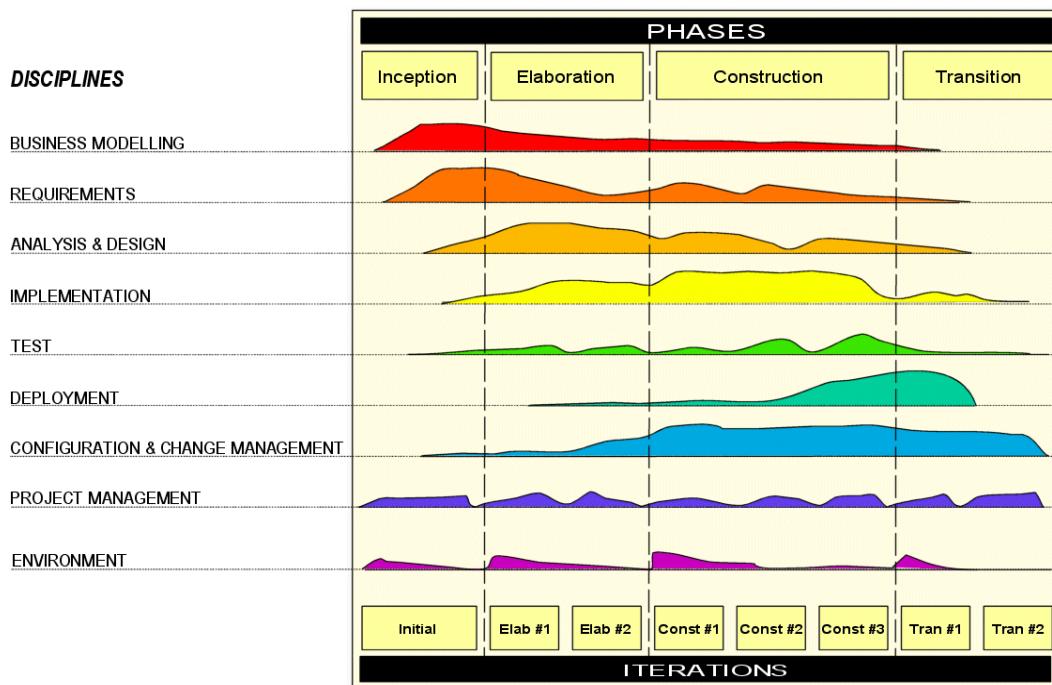
The goals of the elaboration phase are to develop an understanding of the problem domain, establish an architectural framework for the system, and develop the project plan and identify key project risks [10].

- **Construction Phase**

The construction phase is essentially concerned with system design, programming and testing. Parts of the system are developed in parallel and integrated during this phase [10].

- **Transition Phase**

The final phase of the RUP is concerned with moving the system from the development community to the user community and making it work in all environments [10].



*Figure 3.1 Rational Unified Process*

### 3.3 Object Oriented Analysis and Design

Object-oriented analysis and design (OOAD) is a popular technical approach for analysing and designing an application, system, or business by applying object-oriented programming, as well as using visual modelling throughout the development life cycles to foster better stakeholder communication and product quality [11].

### 3.3.1 High Level Use Case Diagram for the Results Management System

The high level use case diagram for the proposed system is depicted in figure 3.2

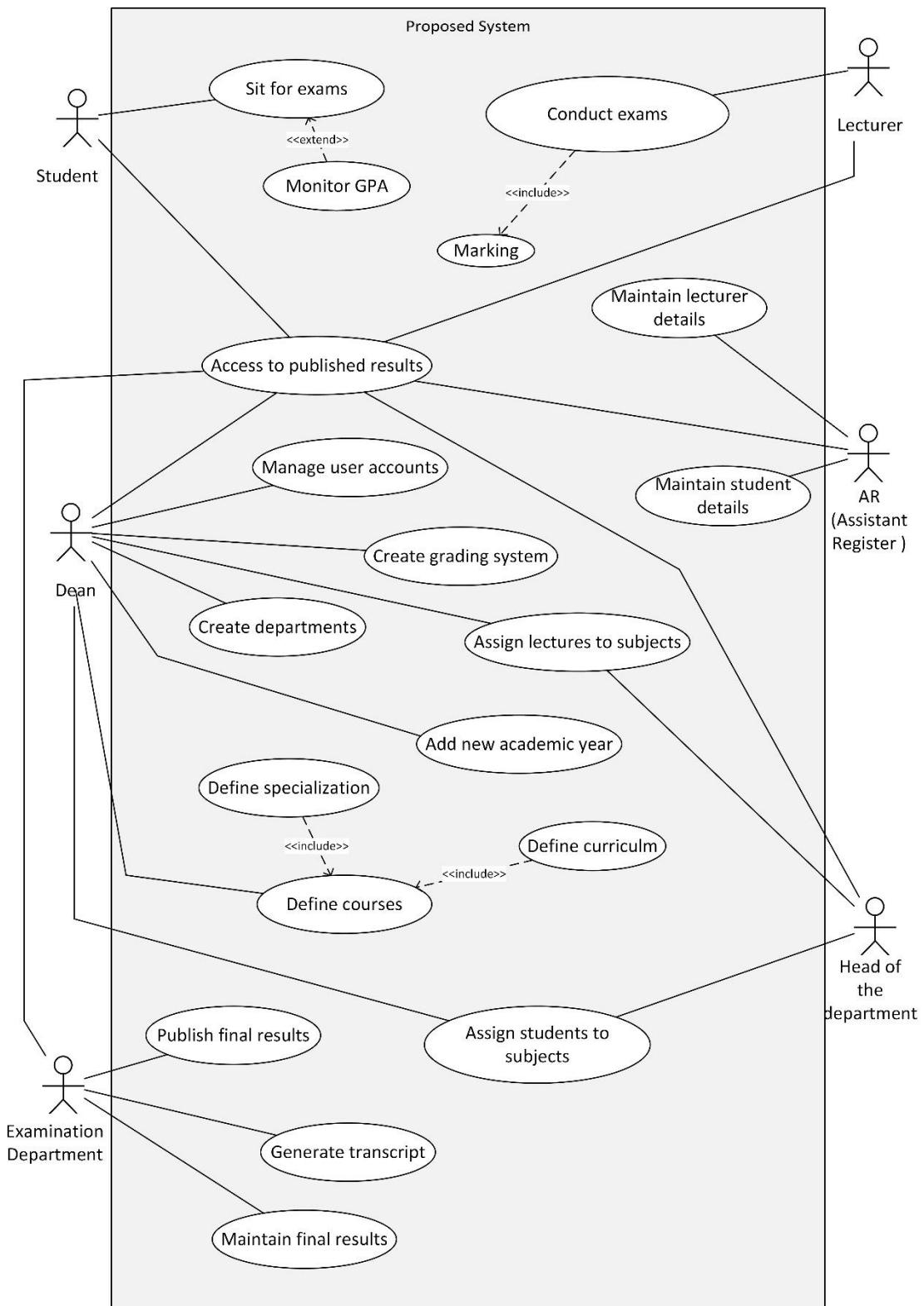


Figure 3.2 High Level Use Case Diagram

### 3.3.2 Class Diagram for the Results Management System

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages [12].

Following figure 3.3 describes the class diagram of the system.

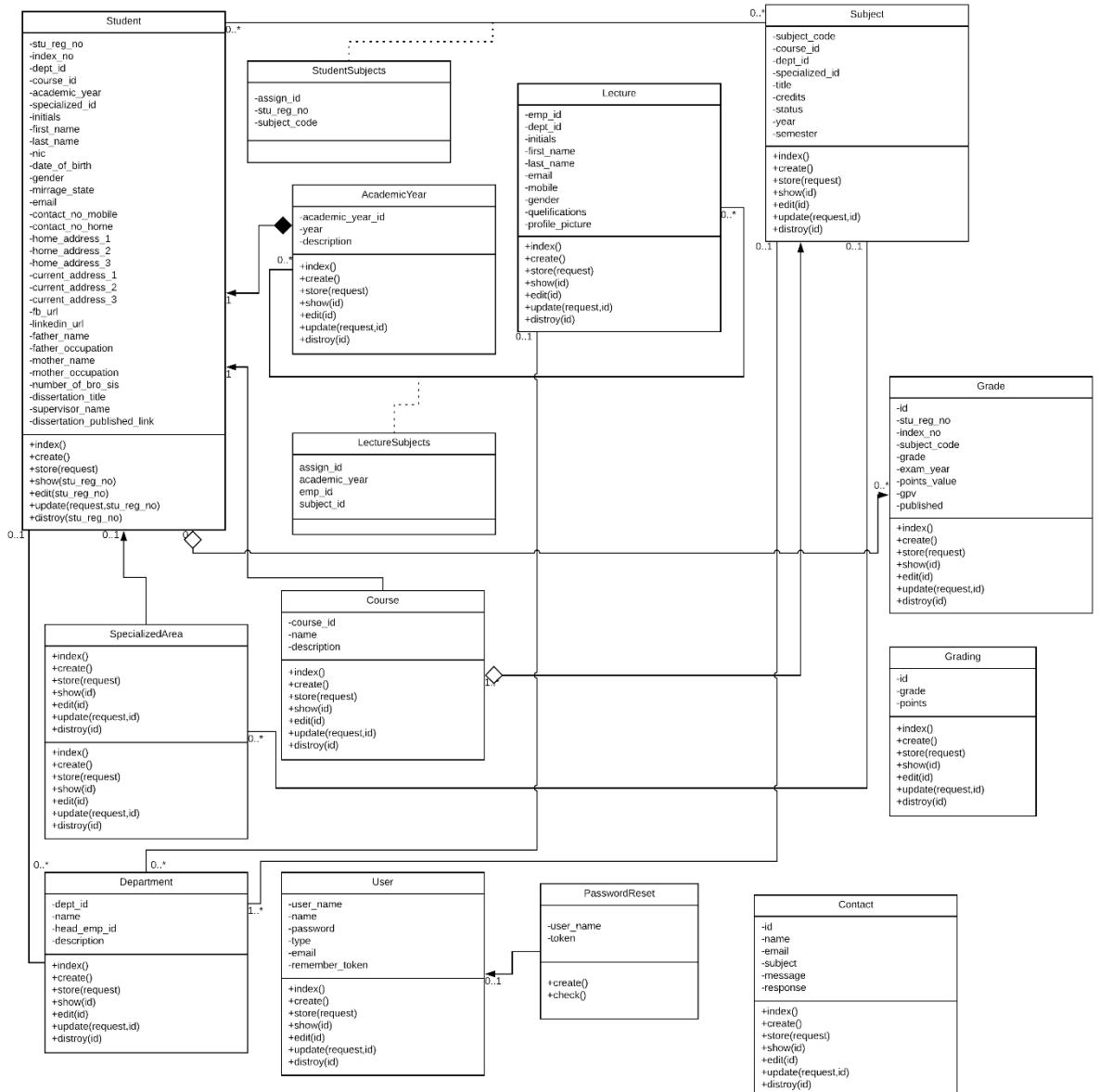


Figure 3.3 Class Diagram

### 3.4 Database Design

Database is an organized collection of data. It is a very important component in information system because it stores all data used by the system. Hence database design should follow techniques such as database normalization to have well organized database. Figure 3.4 describes database design of proposed system.

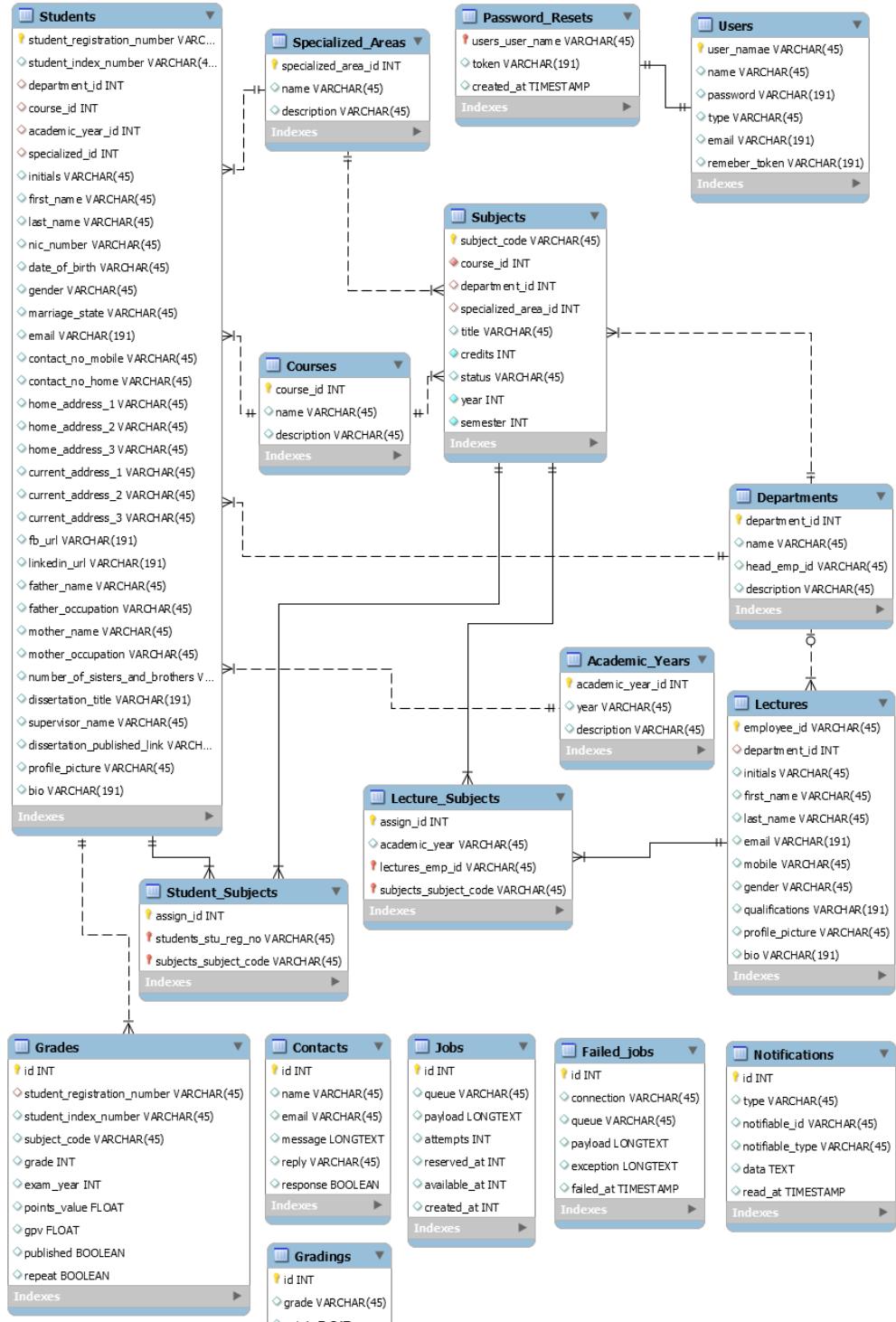


Figure 3.4 Database Design

## 3.5 User Interface Design

User interface design generally refers to the visual layout of the elements that a user might interact with in a computer system. User interface design is a very important part of design phase because interfaces are the component which makes interaction between users and computer system. User interface design focuses on anticipating what users might need to do and ensuring that the interface has elements that are easy to access, understand, and use to facilitate those actions.

### 3.5.1 Key Design Considerations Used for the System

Following design consideration has been taken while designing the system.

- Clean, simple and light interfaces were used by the system to make user friendly interfaces which may take less cognitive energy of users.
- Tab indexing were used for easy navigation between components.
- Meaningful names, tooltips and colours used for relevant components specially for buttons and links.
- Bootstrap [13] panels were used for separate different contents for easy understand ability.
- Flash messages (disappears after refresh) and Toastr [14] notifications were used according to situations to provide feedback to users.

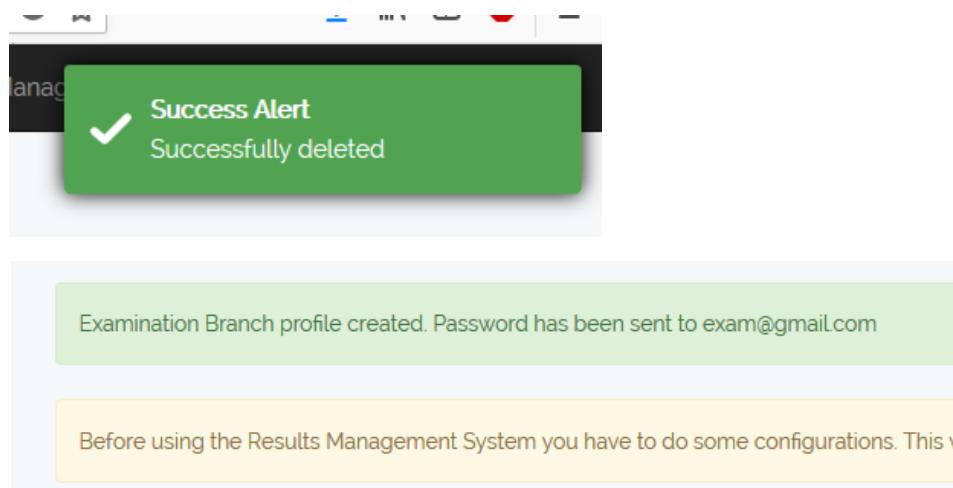


Figure 3.5 Notifications

- Bootstrap modals [15] used to increase user experience and reduce load separate pages.

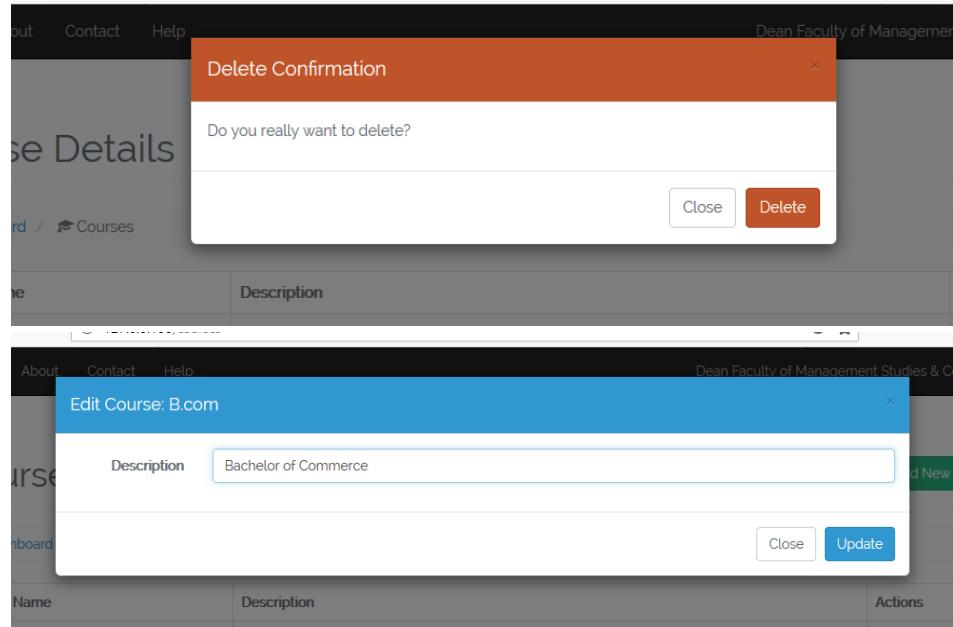


Figure 3.6 Modal Pop-up Windows

- System has provided meaningful error messages to guide users to prevent from errors.

The image displays four separate input fields with their corresponding error messages:

- Password**: The password must be at least 6 characters.
- User Name**: Students should use student registration number as a user name. Ex: 2014c015  
These credentials do not match our records.
- Registration No: \***: Ex:2014c015  
The student registration number has already been taken.
- NIC \***: 9423404V  
The nic number must be at least 10 characters.

Figure 3.7 Meaningful Error Messages

- Dropdowns were used in appropriate places to guide users to prevent from entering wrong inputs.
- Since web-based system, it used responsive development design to become more user friendly in various devices.
- System using validations to prevent using weak passwords to establish security of the system.

The password must be at least 6 characters.

*Figure 3.8 Password Validation*

- System using dynamic input fields to increase user experiences to appropriate places.

*Figure 3.9 Dynamic Components*

- To prevent from misunderstanding of input fields system has given examples in help block.

User Name

Students should use student registration number as a user name. Ex: 2014c015

Registration No:

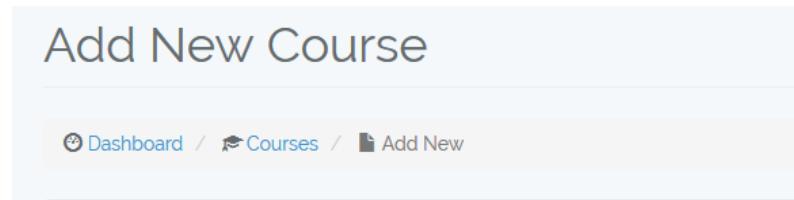
Registration Number

Ex:2014c015

*Figure 3.10 Help Blocks*

- Prevent users from forgetting required input fields system is using required notation, required HTML attribute and tooltips.
- System has top navigation and side navigation bar to easy navigation for users.

- System has provided breadcrumb to understand users where are they and also for easy navigation.

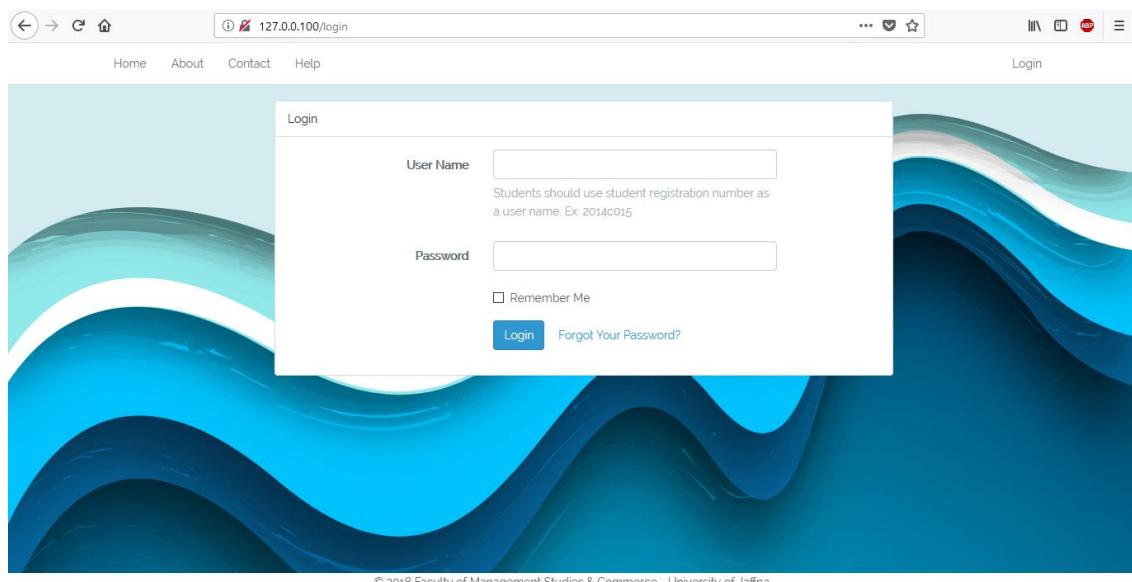


*Figure 3.11 Breadcrumb*

### 3.5.2 Login Interface

This is more general and very important interface in all systems which may use authentication process. Simple and light login interface provided by the system to easy understanding of interface and maintain professional looking interfaces.

All the users can use same login interface to gain access to the system. And according to the provided credentials system redirect to correct dashboard. With additional feature forget password, users may get email message with one-time password reset link. Following figure 3.12 shows login interface of the system.



*Figure 3.12 Login Interface*

### 3.5.3 Configuration Module Interfaces

Configuration module has been developed to configure the system in the first run. This may provide user guided interfaces to get configuration works done without any technical support.

Through this module user can create ‘Dean’ profile and dean of the faculty can continue further configurations such as other admin user profile creation, course creation, setting department, establish grading system etc. This may involve several interfaces in the figure 3.13 shows the initial stage of configuration process called dean profile creation as a super admin of the system.

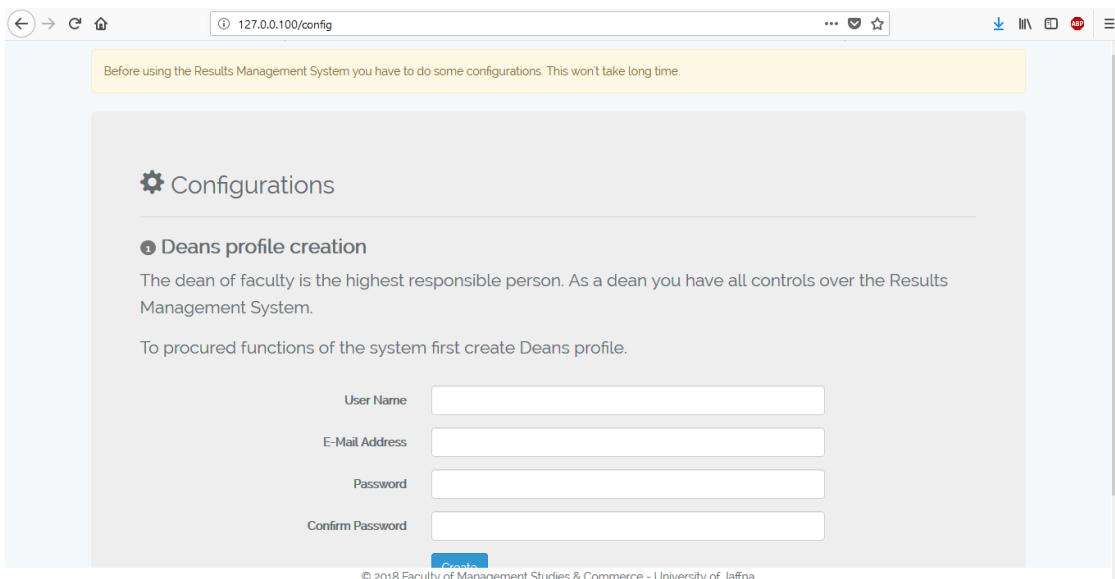


Figure 3.13 Dean Profile Creation - Configuration Interface

### 3.5.4 Dashboard Interfaces

Dashboard is a place which provide navigations to every places and provide summary of the information. The system will provide different dashboard according to the user types. Following figure 3.14 shows ‘Dean’ profile dashboard who is the super user of system.

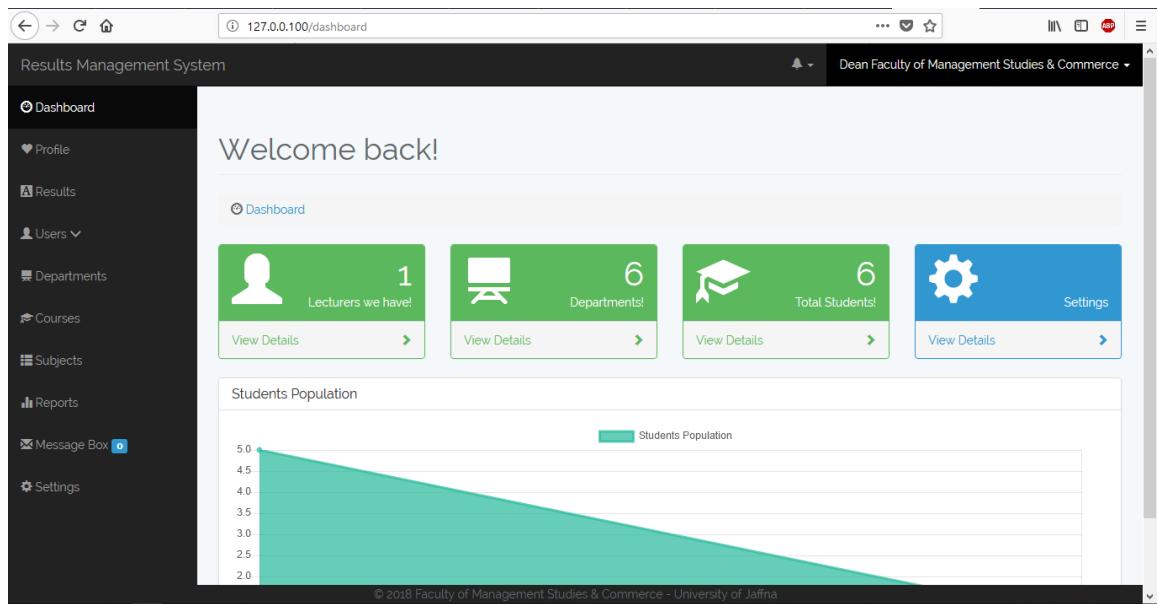


Figure 3.14 Dashboard of Dean

# Chapter 4 - Implementation

## 4.1 Introduction

The objective of the implementation phase is first to install the system in the production environment and to bring it into operation and second, to ensure that the system as developed according to the design phase. This is an important stage in the software development lifecycle, because this stage builds the system into a working system.

This chapter mainly discusses about the implementation environment, the tools and techniques as well as the reusable components used to implement the system, major codes and also module structure.

The system has been developed using modular programming which is software design technique that emphasizes separating the functionality of a program into independent, interchangeable modules, such that each contains everything necessary to execute only one aspect of the desired functionality. Comments were used in proper way to describe the code segments for the future references.

## 4.2 Implementation Environment

The implementation environment can describe according to the hardware and software environment. The following table 4.1 describes the implementation environment of the system.

Hardware Environment	Software Environment
Intel(R) Core(TM) i5-2410M @ CPU 2.30 GHz	Windows 10 Pro
4GB RAM	PHP Version 7.1.9
500GB Hard Disk	MySQL Version 5.7.19
	Apache Version 2.4.27

Table 4.1 Implementation Environment

The system has been developed as a web-based system which can be used as platform independent.

## 4.3 System Development Tools and Technologies

In this section describes tools and technologies used to develop the system. This will cover the used languages, IDE, frameworks and graphic designing tools, etc.

### 4.3.1 Visual Studio Code

Visual Studio Code is a source code editor developed by Microsoft for Windows, Linux and macOS. It includes support for debugging, embedded Git control, syntax highlighting, intelligent code completion, snippets, and code refactoring. This were used as primary IDE for developing [18].

### 4.3.2 PHP

Hypertext Preprocessor is a server-side scripting language designed for Web development, but also used as a general-purpose programming language. It was originally created by Rasmus Lerdorf in 1994, the PHP reference implementation is now produced by The PHP Group.

PHP version 7.1.9 were used as the primary language for the development of the system [19].

### 4.3.3 Laravel

Laravel is a free, open-source PHP web framework, created by Taylor Otwell and intended for the development of web applications following the model–view–controller architectural pattern [17] and based on Symfony.

Laravel version 5.5 were used as the framework to developed the system because of the its features, security, huge community and less learning curve [16].

#### 4.3.4 JavaScript

JavaScript, often abbreviated as JS, is a high-level, interpreted programming language. It is a language which is also characterized as dynamic, weakly typed, prototype-based and multi-paradigm [20].

#### 4.3.5 JQuery

JQuery is a cross-platform JavaScript library designed to simplify the client-side scripting of HTML. It is free, open-source software using the permissive MIT License [21].

#### 4.3.6 Ajax

AJAX is a technique for creating fast and dynamic web pages. AJAX allows web pages to be updated asynchronously by exchanging small amounts of data with the server behind the scenes [22].

#### 4.3.7 CSS

Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a mark-up language like HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript [23].

#### 4.3.8 Bootstrap

Bootstrap is a free and open-source front-end framework for designing websites and web applications. It contains HTML- and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions. Bootstrap version 3 were used in this system [13].

#### **4.3.9 MySQL**

MySQL is a freely available open source Relational Database Management System (RDBMS) that uses Structured Query Language (SQL). SQL is the most popular language for adding, accessing and managing content in a database. It is most noted for its quick processing, proven reliability, ease and flexibility of use. MySQL version 5.7 were used to manage database [24].

#### **4.3.10 Apache HTTP Server**

The Apache HTTP Server, colloquially called Apache, is a free and open-source cross-platform web server, released under the terms of Apache License 2.0. Apache is developed and maintained by an open community of developers under the auspices of the Apache Software Foundation [25].

#### **4.3.11 Adobe Photoshop CC 2018**

Adobe Photoshop is a raster graphics editor developed and published by Adobe Systems for macOS and Windows. Photoshop were used to retouch and edit graphic resources [26].

## 4.4 Major Code Segments

The main code modules developed in the system has been mentioned bellow by briefly describing their functionality. Comments were added in suitable places to identify the functionality of relevant code lines.

### 4.4.1 Middleware Attachment

Several middleware's were used to access control and handle configuration functionalities. Middleware provide a convenient mechanism for filtering HTTP requests entering to the application.

```
public function __construct()
{
    //Set configuration middleware
    $this->middleware('config');

    //Set authentication middleware
    $this->middleware('auth');

    //Set middleware which can accessed by dean & ar only
    //Except index, show functions
    $this->middleware('ARorDean', ['except'=> ['index', 'show']]);
}
```

### 4.4.2 Validations

Laravel's '*ValidatesRequests*' trait was used to validate incoming HTTP request with a variety of powerful validation rules. If the validation rules pass, code will keep executing normally; however, if validation fails, an exception will be thrown and the proper error response will automatically be sent back to the user.

Following code segment were used required, max, string and email validation rules to ensure users inputs.

```

public function store(Request $request)
{
    //Validation rules
    $this->validate($request, [
        'name' => 'required|string|max:255',
        'email' => 'required|string|email|max:255',
        'subject' => 'required|string|max:255',
        'message' => 'required|string',
    ]);

    //Create new message
    $contact = new Contact;
    $contact->name = $request->input('name');
    $contact->email = $request->input('email');
    $contact->subject = $request->input('subject');
    $contact->message = $request->input('message');
    //Save new message
    $contact->save();

    //Return location with notification
    return redirect('/contact/create')->with('success', "Thank
you for your valued feedback.");
}

```

#### 4.4.3 GPA Calculation Trait

This is very important code segment in grades module. This were used to calculate overall as well as semester wise GPA of particular undergraduate. Here the system has to check grading points according to their grades as well as whether the following grade is achieved in proper sitting or it is repeated grade. And then filter out correct subjects to calculate correct GPA. And also when creating semester wise GPA system has to filter out relevant subjects of each semester according to the published results.

The formula for GPA calculation is given below [3].

$$\text{GPA} = \frac{\sum_{i=1}^N (\text{Credit Hours}) \times (\text{Grade Point per Credit})}{\sum_{i=1}^N (\text{Credit Hours})}$$

$$\text{GPA} = \frac{\sum_{i=1}^N (\text{Credit Hours}) \times (\text{Grade Point per Credit})}{\sum_{i=1}^N (\text{Credit Hours})}$$

Where i = Course number and N = the total number of courses considered

Figure 4.1 Formula for GPA calculation

```

//Manage Not Repeated subjects
    //max_points - To identify fresh sitting
    //status      - To calculate GPA

    //Manage max_points
    for($i = 0; count($array) > $i; $i++) {

        //run only for first element
        if($i == 0){
            //if the grade AB || MC || DFR --> max_points(4.0)
            if($array[$i]['1'] == $AB->id || $array[$i]['1'] ==
$MC->id || $array[$i]['1'] == $DFR->id ){
                //max_points(4.0)->[4]
                $array[$i]['4'] = '4.0';
                //status[5]
                $array[$i]['5'] = '0';

            }else{
                //max_points(3.0)->[4]
                //if current points_value > 3.0
                $array[$i]['4'] = '3.0';
                //status[5] --> current points_value
                $array[$i]['5'] = $array[$i]['2'];

            }

            //continue the loop with in the if
            continue;
        }

        //if the current grade is a MC||AB||DFR
        //if previous max_points has 4.0 && current grade is
        acceptable for max_value(4.0)
        if($array[$i]['1'] == $AB->id || $array[$i]['1'] == $MC-
>id || $array[$i]['1'] == $DFR->id ){
            //Check previous max_points[4] & Status[5]
            if($array[$i - 1]['4'] == '4.0' && $array[$i -
1]['5'] == '0'){
                //It means fresh sitting
                //max_points(4.0)->[4]
                $array[$i]['4'] = '4.0';
                //status[5] --> get from previous status
                $array[$i]['5'] = '0';

            }else{
                //It means previous one is MC|AB|DFR but not a fresh
                sitting
                //max_points(3.0)->[4]
                $array[$i]['4'] = '3.0';

                //Manage status
                //points_value[2]
                //status[5]
                //if equal or below status[4]
                if($array[$i - 1]['5'] <= '3.0'){
                    //status[5]
                    $array[$i]['5'] = $array[$i - 1]['5'];
                }else{
                    //status[5] ----> Should be '3.0'
                    $array[$i]['5'] = '3.0';
                }

            }
        }
    }
}

```

```

else{
    //if current grade is not MC || AB || DFR

        // if previous one is a MC||AB||DFR
        if($array[$i-1]['1'] == $AB->id || $array[$i-1]['1']
== $MC->id || $array[$i-1]['1'] == $DFR->id ){
            //if it is a fresh sitting
            if($array[$i - 1]['4'] == '4.0' && $array[$i -
1]['5'] == '0') {
                //max_points(4.0)->[4]
                $array[$i]['4'] = '4.0';
                //status[5] -->set from current points_value
                $array[$i]['5'] = $array[$i]['2'];
            }else{
                //But not a fresh sitting
                //max_points(4.0)->[4]
                $array[$i]['4'] = '3.0';

                //Manage status
                //points_value[2]
                //Should be equal or below current
                max_points[4]
                // if current points_value < previous statue
                if($array[$i]['2'] <= $array[$i - 1]['5'] ) {
                    //get previous state
                    $array[$i]['5'] = $array[$i - 1]['5'];
                }else{
                    //assign current points_value with 3.0
                    condition
                    if($array[$i]['2'] <= '3.0'){
                        //status[5]
                        $array[$i]['5'] = $array[$i]['2'];
                    }else{
                        //status[5] ----> Should be '3.0'
                        $array[$i]['5'] = '3.0';
                    }
                }
            }
        }else{
            //if previous one is not a MC||AB||DFR

                //max_points(3.0)->[4]
                $array[$i]['4'] = '3.0';

                //Manage state
                //points_value[2]
                if($array[$i]['2'] <= $array[$i-1]['5'] ) {
                    //state[5]
                    $array[$i]['5'] = $array[$i-1]['5'];
                }else{
                    //Should be equal or below max_points[4]
                    if($array[$i]['2'] <= '3.0'){
                        //state[5]
                        $array[$i]['5'] = $array[$i]['2'];
                    }else{
                        //state[5] ----> Should be '3.0'
                        $array[$i]['5'] = '3.0';
                    }
                }
            }
        }
    } //End of for loop
}

```

Above code segment shows the calculation of GPA of repeated subject in different exam years. It used array as well as number of for-loops to filter out correct subjects with credit values and grade point values. Here as the first stage filter out repeated subjects using ‘repeat’ attribute. And then ‘max\_points’ and ‘status’ variables were used to identify correct points value.

```
//Manage Not Repeated subjects

//Find SUM of gpv
$gpv += DB::table('grades')
->where([['student_registration_number', '=', $id],
['published', '=', true], ['repeat', '=', false]])
->get()
->sum('gpv');

//Find SUM of credits
$credits += DB::table('grades')
->where([['student_registration_number', '=', $id],
['published', '=', true], ['repeat', '=', false]])
->get()
->sum('credits');
```

When comparing to the GPA calculation of not repeated which means proper sitting exams are very easy to identify and calculate assigned GPA. Above code segment shows the calculation GPA of not repeated subject exams. This were used ‘repeat’ and ‘published’ attributes to identify correct subjects.

#### 4.4.4 Student Model

The Eloquent ORM included with Laravel provides a beautiful, simple active record implementation for working with the database. Each database table has a corresponding ‘Model’ which is used to interact with that table. Models allow to query for data in tables, as well as insert new records into the table.

The following code segment shows ‘Student’ model. It describes the different relationships exist in the system and relevant configurations. Student model were used to manage undergraduate details of the faculty.

Here it has overridden primary key as ‘student\_registration\_number’ and also disabled incrementing feature of primary key.

```

<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

//Student model
class Student extends Model
{
    //primary key field
    protected $primaryKey = 'student_registration_number';
    //Set primary key incrementing false
    public $incrementing = false;

    //Relationship with Grades
    public function grade(){
        // Foreign key:student_registration_number
        return $this-
    >hasMany('App\Grade', 'student_registration_number');
    }

    //Relationship with SpecializedArea
    public function specializedArea(){
        // Foreign key:specialized_area_id
        return $this-
    >belongsTo('App\SpecializedArea', 'specialized_area_id');
    }

    //Relationship with courses
    public function course(){
        // Foreign key:course_id
        return $this->belongsTo('App\Course', 'course_id');
    }

    //Relationship with Department
    public function department(){
        // Foreign key:department_id
        return $this->belongsTo('App\Department', 'department_id');
    }

    //Relationship with AcademicYear
    public function academicYear(){
        // Foreign key:academic_year_id
        return $this-
    >belongsTo('App\AcademicYear', 'academic_year_id');
    }

    //Relationship with subjects
    public function subject(){
        // students_stu_reg_no
        // subjects_subject_code
        return $this-
    >belongsToMany('App\Subject', 'student_subjects', 'students_stu_reg_no',
        'subjects_subject_code');
    }
}

```

#### 4.4.5 Delete Function via Ajax

The following code segment describes the deletion functionality created using Ajax [22]. Here Toastr [14] notifications were used to give feedback to user.

```
// delete function
$(document).on('click', '.delete-modal', function() {
    $('#deleteModal').modal('show');
    id = $(this).data('id');
});
$('.modal-footer').on('click', '.delete', function() {
    $.ajax({
        type: 'DELETE',
        url: 'departments/' + id,
        data: {
            '_token': $('input[name=_token]').val(),
        },
        success: function(data) {
            toastr.success('Successfully deleted', 'Success Alert', {timeOut: 5000});

            //Remove table row
            $('.item' + id).remove();
        }
    });
});
```

#### 4.4.6 Send Mail Job

Queues allow to defer the processing of a time consuming task, such as sending an email, until a later time. Deferring these time consuming tasks drastically speeds up web requests to the application. The following code segment shows the job class for send default password through the queue.

And also for the customizing purpose this were used ‘Mailable’ feature which comes with Laravel [16] by default.

```

<?php

namespace App\Jobs;

use Illuminate\Bus\Queueable;
use Illuminate\Queue\SerializesModels;
use Illuminate\Queue\InteractsWithQueue;
use Illuminate\Contracts\Queue\ShouldQueue;
use Illuminate\Foundation\Bus\Dispatchable;

//For Sending mail
use Illuminate\Support\Facades\Mail;

//Mailable file
use App\Mail\sendMailMailable;

// This is for sending default password to AR & Exam Branch &
Lectures

class SendEmailJob implements ShouldQueue
{
    use Dispatchable, InteractsWithQueue, Queueable,
SerializesModels;

    /**
     * Create a new job instance.
     *
     * @return void
     */

    //used to store construct values
protected $request;

public function __construct($request)
{
    $this->request = $request;
}

/**
 * Execute the job.
 *
 * @return void
 */
public function handle()
{
    Mail::to($this->request['email'])->queue(new
sendMailMailable($this->request));
}
}

```

## 4.5 Re-used Components

The following re-usable components have been used when implementing the system to add more attractiveness and to maximize the efficiency of the system.

### 4.5.1 Toastr - JavaScript Notification

Toastr is a Javascript library for Gnome / Growl type non-blocking notifications. jQuery is required. The goal is to create a simple core library that can be customized and extended. Toastr is under MIT license [14].

### 4.5.2 Bootstrap select

The jQuery plugin that brings searchable dropdowns to the system. Bootstrap-select requires jQuery v1.9.1+, Bootstrap's dropdown.js component, and Bootstrap's CSS. This plugin may increase the user experience through ease of use [27].

### 4.5.3 Laravel-DOM pdf

Dompdf is a HTML layout and rendering engine written in PHP. Laravel-DOM pdf is Laravel way of using this component. This package were used to generate pdf documents in this system [28].

### 4.5.4 Laravel Excel

Laravel Excel is intended at being Laravel-flavoured PhpSpreadsheet: a simple, but elegant wrapper around PhpSpreadsheet with the goal of simplifying exports. This component was used to import and export excel files [29].

#### **4.5.5 Laravel-chartjs**

Chart.js is a JavaScript library that allows you to draw different types of charts by using the HTML5 canvas element. Laravel-chartjs is a wrapper for laravel of chartjs [30].

#### **4.5.6 SB Admin – Dashboard Template**

SB Admin bootstrap 3 admin template were used as dashboard of the system. This template was customized according to the needed requirements of the system [31].

### **4.6 Module Structure**

Main modules developed in the system and its brief description of functionality are listed below.

- Configuration Module

This module is responsible for checking and providing suitable user interfaces to the authorized users for manage configurations in the system.

- Notification Module

This is the module which manages relevant notifications to users. Database connection is used to manage various notifications among the users.

- Settings Module

Settings module is a sensitive section of the system because it manages core functionality of the system. Some settings are very critical to the main functions.

- Academic Years Module

Academic Year is a very important factor for the academic activities in the system. This may handle academic year details of undergraduates.

- Courses Module

Each and every student following a course which is assigned by the relevant authority. Courses module manage courses details conducted by the faculty.

- Dashboard Module

There are several types of users are involving with the system. Namely, undergraduates, lectures, dean, assistant register, examination branch.

Dashboard module take care of providing correct interface according to the authentication.

- Departments Module

Normally a faculty having several departments which may specialized with different academic areas. This module is responsible for managing department details of the faculty.

- Authentication Module

This module is responsible for authenticate users, password recovery and registration of new users to the system. Authentication module is built in feature which comes with Laravel [16]. This module was customized to match with the system requirements. For example, login with username instead of email, user registrations process restricted to authorized users, sending random password through email.

- Grades Module

This is the important section of the system because grades module manages all the activities related to grades/results of undergraduates.

- Lecturers Module

Lecturers module manage all the activities related to lecturers. For example, lecturers' details, enrolled subjects, etc.

- Students Module

This module is responsible for managing all the details of undergraduates.

- **Reports Module**

Reports module is responsible for generating different reports according to the user request in the form of pdf or excel. For example, semester results, full details of an undergraduate.

- **Subjects Module**

This module manages the subject details of each course conducted by faculty.

# **Chapter 5 - Evaluation**

## **5.1 Introduction**

“A software evaluation is a type of assessment that seeks to determine if software or a combination of software programs is the best possible fit for the needs of a given client. The idea is to look closely at the resources and tools provided by the software that is either currently in use or is being examined as a possible addition to programs already in use by that client. Based on a prepared list of criteria along with some practical experimentation, a software evaluation makes it possible to determine if the products would be helpful to the client or if some other combination of software products would serve to better advantage” [32].

## **5.2 System Test Plan and Test Cases**

Planning a test case is a very important aspect for developing system as well as for the completed system. Test plan should have the ability to test the functionality of the overall system. By properly testing a system, it can identify the errors which generate from the system and can correct them. The implemented system was tested using different test cases.

Since the development started, the test plan continued by testing the system units. After completing a system unit, it was completely tested to identify whether it can function according to expectations. Therefore, early detection of the errors was helped by this testing stage.

After performing the system unit testing, next integration testing was done, and this can identify the errors and the required functionality of the units after integration. System testing was performed as the final stage for the completely developed web based application to check the functionality. In order to reduce the complexity of the system, test cases were written for each module. Following Table 5.1 shows the test plan and please refer Appendix E for other test cases.

<b>Module Name</b>	<b>Function Name</b>	<b>Test Priority</b>
<b>Configuration</b> <b>Module</b>	Dean's profile creation	High
	AR's profile creation	High
	Exam branch profile creation	High
	Other configurations	Medium
<b>Settings</b> <b>Module</b>	Assign specialization to students	High
	Add specialization subjects to student	High
	Update specialization subjects to student	High
	Lecturers - Subjects Enrolment	Medium
	Delete enrolled subjects of a lecturer	Medium
	Add specialized areas of the faculty	High
	Edit specialized areas of the faculty	High
	Delete specialized areas of the faculty	High
	Add academic years	High
	Edit academic years	High
	Delete academic years	High
	Add grading points	High
	Edit grading points	High
	Delete grading points	High
<b>Notifications</b> <b>Module</b>	Send new result publish alert to students	High
	System notification of new results to students	High
<b>Courses</b> <b>Module</b>	Add new course	High
	Edit course details	High
	Delete a course	High
<b>Departments</b> <b>Module</b>	Add new department	High
	Edit existing department	High
	Delete department	High
<b>Authentication</b> <b>Module</b>	Login user	High
	Provide correct dashboard	High
	Register user	High
	Send default password	High
	Recover forget password	High
	Manage too many login attempts	Medium

<b>Profiles</b>	Change password	High
<b>Module</b>	Edit user details	High
<b>Grades</b> <b>Module</b>	Add new grades	High
	Edit grades	High
	Publish grades	High
	Add repeat subject grades	High
	Search by student registration number	Medium
	Search by student index number	Medium
	Search by student name	Medium
	Search by subject	Medium
	Calculate overall GPA	High
	Calculate semester wise GPA	High
<b>Lecturers</b> <b>Module</b>	Draw GPA progress line chart	Medium
	List down relevant subjects with grade of each undergraduate	High
	Add new lecturer	High
	Edit existing lecturer	High
	Search lecturer	Medium
<b>Students</b> <b>Module</b>	Export showing list of lecturers	Medium
	Add new student	High
	Edit existing student	High
	Search student	Medium
	Export showing list of undergraduates	Medium
<b>Reports</b> <b>Module</b>	Students in specific academic year	Medium
	Students in specific course	Medium
	Students in specific department	Medium
	Students in specific specialized area	Medium
	Full details of a student	High
	Assigned subjects of a student	High
	All the lecturers in the faculty	Medium
	Lecturers in specific department	Medium
	Assigned subjects of specific lecturer	Medium
	Full results of a student	High

	Subject results	Medium
	Semester results	High
	Course details	Medium
	Grading details	Medium
<b>Subjects</b>	Add new subject	High
	Edit existing subject	High
	Delete existing subject	High

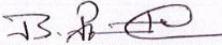
*Table 5.1 High level test plan*

### 5.3 User Acceptance Testing

After implementing the system, it was tested in the client environment to get the user acceptance testing. The system was tested by the client to identify the functionality provided by the system, whether it can satisfy the operational needs of the faculty and examination branch. User Acceptance Testing was started by feeding the actual dataset to the developed system.

User acceptance test has been done by the dean as the top level user as well as other several users. After completing the user acceptance testing, a positive response was received from all the users.

User feedbacks were collected using questioner form to evaluate the system. Following figure 5.1 shows one of completed evaluation forms in role of lecturer.

Name of User : Mr.B.Prahathan		Role of User : Lecturer			
Evaluating Item	Very Good	Good	Average	Poor	Very Poor
Ease of use	✓				
System navigation	✓				
Functionalities	✓				
User friendliness	✓				
Ease of learning	✓				
Response time	✓				
Security	✓				
Colour scheme	✓				
Overall performance	✓				
Design	✓				
Comments	To be checked with expert and then only implement				
Date:	28/9/2018				
Signature: ..... 					

HEAD  
 Department of Commerce  
 Faculty of Mgt. Studies & Commerce  
 University of Jaffna  
 Sri Lanka

Figure 5.1 User evaluation questionnaire - Lecturer

Following figure 5.2 shows one of completed evaluation forms in role of student.

User Evaluation Questionnaire					
Name of User:	D.T.K. Jayasekara		Role of User: Student		
Evaluating Item	Very Good	Good	Average	Poor	Very Poor
Ease of use	<input checked="" type="checkbox"/>				
System navigation	<input checked="" type="checkbox"/>				
Functionalities	<input checked="" type="checkbox"/>				
User friendliness	<input checked="" type="checkbox"/>				
Ease of learning	<input checked="" type="checkbox"/>				
Response time	<input checked="" type="checkbox"/>				
Security	<input checked="" type="checkbox"/>				
Colour scheme	<input checked="" type="checkbox"/>				
Overall performance	<input checked="" type="checkbox"/>				
Design	<input checked="" type="checkbox"/>				
Comments					

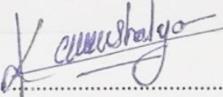
Date: 2018-09-25 Signature: 

Figure 5.2 User evaluation questionnaire - Student

Following figure 5.3 shows one of completed evaluation forms in role of examination branch.

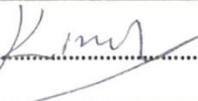
User Evaluation Questionnaire					
Name of User :	R. Kirupan		Role of User : Examination Branch		
Evaluating Item	Very Good	Good	Average	Poor	Very Poor
Ease of use	✓				
System navigation		✓			
Functionalities		✓			
User friendliness	✓				
Ease of learning			✓		
Response time		✓			
Security		✓			
Colour scheme			✓		
Overall performance	✓				
Design		✓			
Comments					
Date:	26.09.2018				
Signature:					

Figure 5.3 User evaluation questionnaire - Examination Branch

The summary of user friendliness in user acceptance test is described in figure 5.4 and overall performance is described in figure 5.5. The certificate received from the client has been appended to Appendix G.

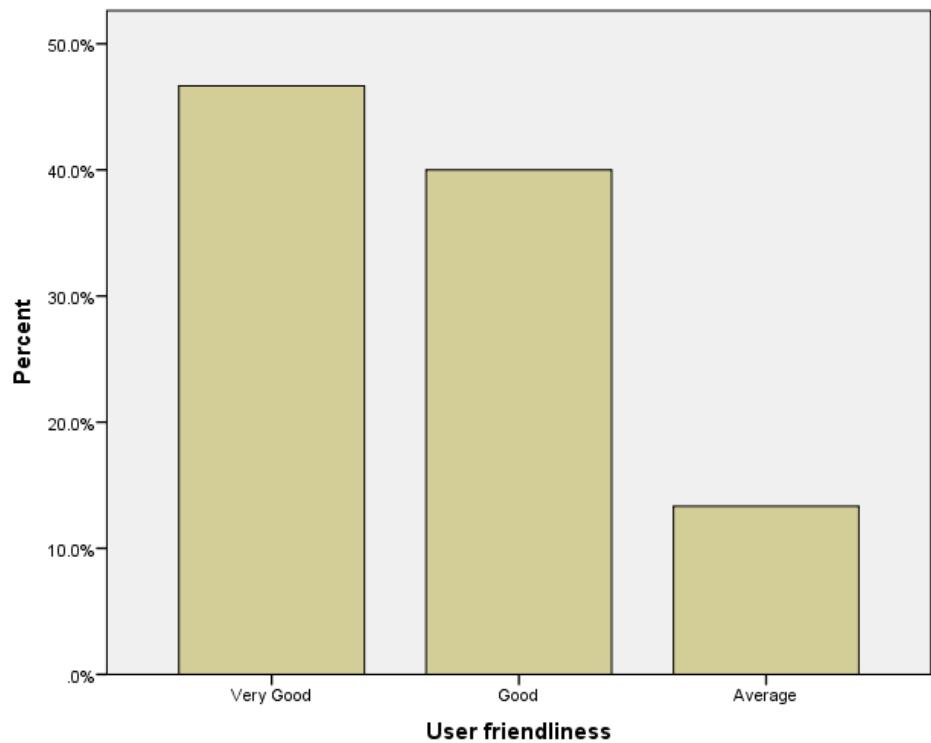


Figure 5.4 Summary of user friendliness

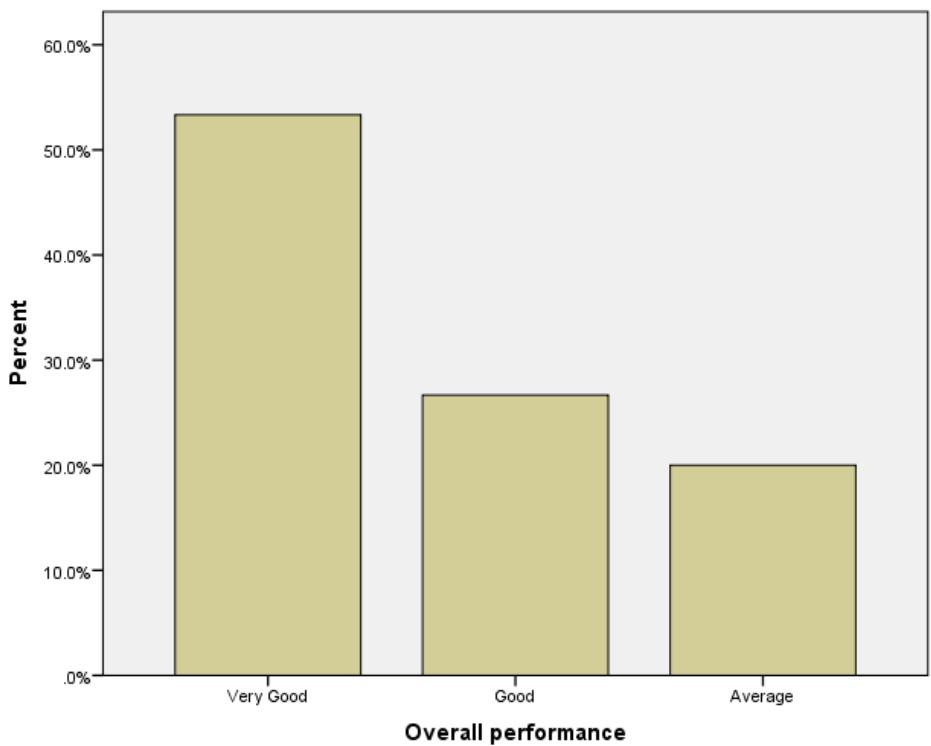


Figure 5.5 Summary of overall performance

# **Chapter 6 - Conclusion**

## **6.1 Introduction**

The Faculty of Management Studies & Commerce is one of the faculties in University of Jaffna which is offering two major undergraduate courses which lead to Bachelor of Business Administration (B.B.A.) and Bachelor of Commerce (B.Com.) degrees with the vision of “To provide best possible human resources to society in the field of Management and Commerce”.

The faculty used manual system to manage student results with the help of examination branch. This was very time consuming and less productive way to publish and managing undergraduate’s results. The critical issue was managing repeating subjects. It means managing not proper sitting exams. Authorized parties have to check each and every subject marks to calculate GPA and result sheets. Managing student details also was a critical issue to the faculty. The students in the faculty do not have any kind of system to track their academic carrier online. Most of the time they have to go to the faculty and manually check the subject results.

Through the proposed “Results Management System” has successfully developed, tested and implemented to overcome above mentioned issues of the faculty with the main objective to provide better experience for undergraduates in the faculty while enhancing operational efficiency of the faculty and examination branch.

By comparing test results, user feedbacks, system functionality, it was identified as a system which can satisfy the client requirements up to a satisfactory level.

## **6.2 Critical Evaluation of the Project**

Following figure 6.1 summarize the user acceptance test which was taken from the 15 respondents. The scale is taken from zero to ten for the evaluation. For an example “Functionalities” factor has taken up to nine points.

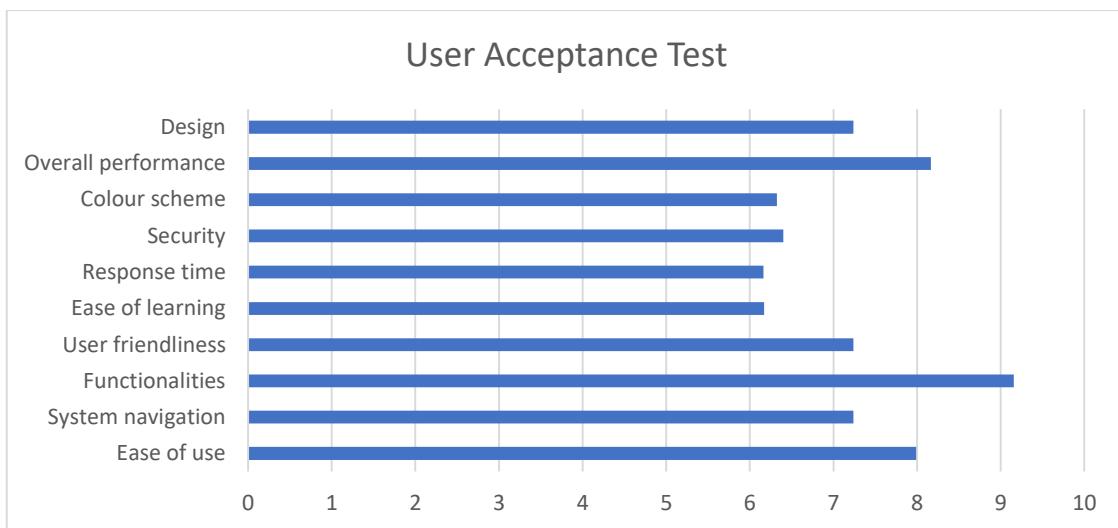


Figure 6.1 Summary of user acceptance test

When comparing with similar systems, both the developed system and similar systems has access for the dean, lecturers, students and also examination branch. But the developed system is highly customized to fit with the specific requirements of the faculty. Some important points are listed below.

- Configuration can be done easily through the configuration module than the similar systems.
- The system is having own GPA calculation algorithm based on the university requirements.
- The process of results publishing is designed according to the university regulations.
- System require lesser learning curve with minimum effort than the other systems.
- Similar systems and the developed system having online results viewer and searching facility.
- The reports are more common and there are some faculty specific reports also included.
- Notification facilities were given as the similar systems.

## 6.3 Future Improvements

Developing a commercial level system is a huge task and very responsible work specially to systems involving sensitive information. Some suggestions for improvements in the future as follows:

- Add more reports to the system

Reports are very important feature to the management level of the system. It is suitable to add new reports to the system which used deeper analysis. For example, trends among various departments, overall student performances in grades with different academic year.

- Improve dean's dashboard

By improving dean's dashboard, it is very easy to monitor various activities in the system. For example, adding new charts to the dashboard.

- Improve the system to match whole university.

By improving several features in current system, it is possible to use for the entire university. This will overcome all bottlenecks having in the university system.

## 6.4 Lessons Learnt

As an undergraduate following a degree program, this was a great opportunity for me to solve practical problem exists in real word using the previously learned lessons. Developing a system to university level, it was a great opportunity to understand university structure and its process.

With the communication of real customers, I have gained new experience specially in analysing their requirements, understand the business domain and negotiation with them. Through the development of web based system I have learnt lot of new technologies and tools specially, I have learnt Laravel [16] from the basic through

development of this system. Apart from technological stuff, I could learn to write reports according to the recognized standard and to present reports to interested parties who wish to read.

I am so grateful to initiate a project to the university which I currently study as an undergraduate.

# References

- [1] Wikipedia, University of Jaffna History [Online]  
Available: [https://en.wikipedia.org/wiki/University\\_of\\_Jaffna](https://en.wikipedia.org/wiki/University_of_Jaffna) [Accessed: 15.04.2018]
- [2] University of Jaffna official web site [Online]  
Available: <http://www.jfn.ac.lk/> [Accessed: 15.04.2018]
- [3] University of Jaffna – Sri Lanka, Faculty of Management Studies & Commerce, Hand Book – Academic Year 2015/2016.
- [4] Search Software Quality, Requirement Analysis [Online]  
Available: <http://searchsoftwarequality.techtarget.com/definition/requirements-analysis> [Accessed: 19.04.2018]
- [5] OpenSIS Pro, Student Information System [Online]  
Available: <http://www.opensis.com/opensis-pro/> [Accessed: 25.04.2018]
- [6] TotalSchool, SolutionDots - School Management System [Online]  
Available: <https://solutiondots.com/school-management-system/> [Accessed: 25.04.2018]
- [7] Wikipedia, Functional requirements [Online]  
Available: [https://en.wikipedia.org/wiki/Functional\\_requirement](https://en.wikipedia.org/wiki/Functional_requirement) [Accessed: 02.05.2018]
- [8] Wikipedia, Non-Functional requirements [Online]  
Available: [https://en.wikipedia.org/wiki/Non-functional\\_requirement](https://en.wikipedia.org/wiki/Non-functional_requirement) [Accessed: 02.05.2018]
- [9] Techopedia, System Design [Online]  
Available: <https://www.techopedia.com/definition/29998/system-design> [Accessed: 20.05.2018]
- [10] Ian Sommerville, Software Engineering, 9<sup>th</sup> ed. 2011
- [11] Wikipedia, Object-oriented analysis and design [Online]  
Available: [https://en.wikipedia.org/wiki/Object-oriented\\_analysis\\_and\\_design](https://en.wikipedia.org/wiki/Object-oriented_analysis_and_design) [Accessed: 01.06.2018]
- [12] Tutorialspoint.com, UML – Class diagram [Online]  
Available: [https://www.tutorialspoint.com/uml/uml\\_class\\_diagram.htm](https://www.tutorialspoint.com/uml/uml_class_diagram.htm) [Accessed: 02.06.2018]

- [13] Bootstrap - Front-end component library [Online]  
Available: <https://getbootstrap.com/> [Accessed: 01.07.2018]
- [14] Toastr - Simple JavaScript toast notifications [Online]  
Available: <https://codeseven.github.io/toastr/> [Accessed: 02.07.2018]
- [15] Modal - Bootstrap's JavaScript modal plugin [Online]  
Available: <https://getbootstrap.com/docs/4.0/components/modal/> [Accessed: 02.07.2018]
- [16] Laravel – PHP Framework [Online]  
Available: <https://laravel.com/> [Accessed: 02.07.2018]
- [17] MVC – Architectural Pattern [Online]  
Available: <https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller> [Accessed: 03.07.2018]
- [18] Visual Studio Code [Online]  
Available: <https://code.visualstudio.com/> [Accessed: 25.08.2018]
- [19] PHP [Online]  
Available: [php.net/](http://php.net/) [Accessed: 25.08.2018]
- [20] JavaScript Tutorial - W3Schools [Online]  
Available: <https://www.w3schools.com/js/> [Accessed: 25.08.2018]
- [21] JQuery [Online]  
Available: <https://jquery.com/> [Accessed: 25.08.2018]
- [22] AJAX Introduction - W3Schools [Online]  
Available: [https://www.w3schools.com/php/php\\_ajax\\_intro.asp](https://www.w3schools.com/php/php_ajax_intro.asp) [Accessed: 26.08.2018]
- [23] CSS Tutorial - W3Schools [Online]  
Available: <https://www.w3schools.com/Css/> [Accessed: 26.08.2018]
- [24] MySQL [Online]  
Available: <https://www.mysql.com/> [Accessed: 26.08.2018]
- [25] The Apache HTTP Server Project [Online]  
Available: <http://httpd.apache.org/> [Accessed: 26.08.2018]
- [26] Photoshop CC [Online]  
Available: <https://www.adobe.com/products/photoshop.html> [Accessed: 29.08.2018]
- [27] Bootstrap Select - JQuery Plugin [Online]  
Available: <https://developer.snapappointments.com/bootstrap-select/>

- [Accessed: 02.09.2018]
- [28] Laravel – DOM pdf [Online]  
Available: <https://github.com/barryvdh/laravel-dompdf>  
[Accessed: 02.09.2018]
- [29] Laravel – Excel [Online]  
Available: <https://laravel-excel.maatwebsite.nl/3.0/getting-started/>  
[Accessed: 02.09.2018]
- [30] Laravel – ChartJs [Online]  
Available: <https://github.com/fxcosta/laravel-chartjs>  
[Accessed: 02.09.2018]
- [31] SB Admin [Online]  
Available: <https://github.com/BlackrockDigital/>  
[Accessed: 05.09.2018]
- [32] Software Evaluation [Online]  
Available: <https://www.wisegeek.com/what-is-software-evaluation.htm>  
[Accessed: 07.09.2018]

# Appendix A – System Manual

This documentation consists of a set of steps to show, how to install this “Results Management System”. Following steps explain about the hardware and software environment which needs to be installed. When installing the system, this documentation can be followed by the interested parties.

- Step one – System Requirements

## Hardware Requirements

Hardware	Minimum Requirements
Processor	3.0 GHz Processor or higher
Memory	1GB
Hard Disk	40GB
Display	1027 x 768 or higher (For better experience)
Internet	Average connection

*Table A.1 Hardware requirements*

## Software Requirements

Software	Minimum Requirements
Operating System	Windows / Linux / MacOS (Platform independent)
Server	WAMP or any other server PHP version 7.1.9 , MySQL version 5.7
IDE	Visual Studio Code / NetBeans (Only for further development)
Web Browser	Firefox / Chrome / Edge or any (Latest updated web browser)

*Table A.2 Software requirements*

- Step two – Installation of Pre-Required Software

### **Server installation – WAMP**

Download and install WAMP server, can be found here:

<http://www.wampserver.com/en/> (Refer to the table A.2 for the versions of dependencies.)

## Installation of web browser

Download and update latest web browser according to the platform. (Refer to the table A.2 for further details.)

- Step three – File Extraction

Open the CD and copy “Results Management System” folder which is inside the “Source” folder and paste it to the “www” folder in WAMP server installation path. File structure of the system shows following figure A.1.

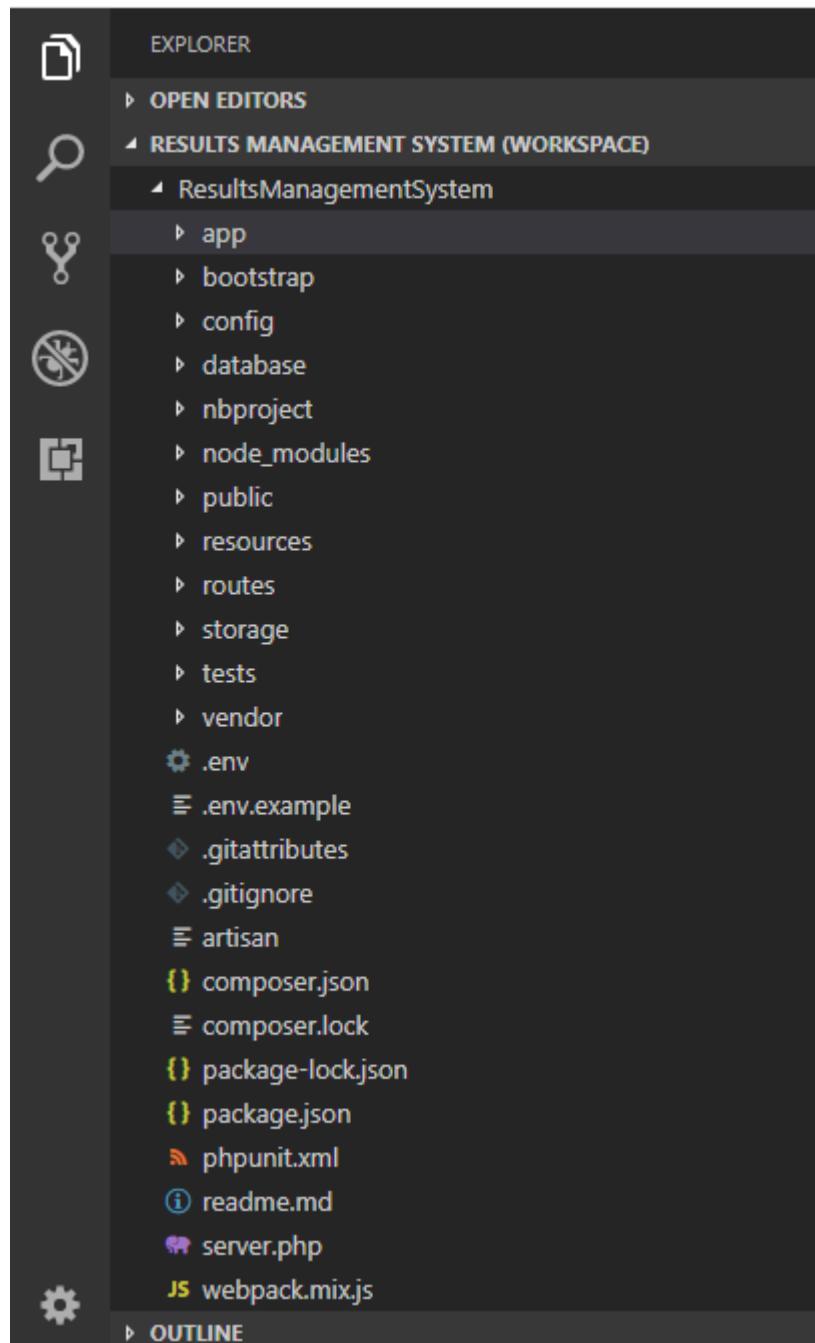


Figure A.1 File structure of the system

- Step four – Database Installation
  1. Open the web browser and type the URL <http://localhost/phpmyadmin/> and enter Username and Password (if you set username and password).
  2. Create empty database by providing name as “rms” and navigate to the “Import” tab and click “Choose file” button. Then browse the CD and select the “rms.sql” file by opening “Database” folder.
  3. Then Press “Go” button located in the bottom of the page.

- Step five – Running the Queue Worker

Laravel [16] includes a queue worker that will process new jobs as they are pushed onto the queue. You may run the worker using the **queue:work** Artisan command. Note that once the **queue:work** command has started, it will continue to run until it is manually stopped or you close your terminal.

Open command prompt/terminal and change directory to the system installed path described in step three. And enter the following command.

`“php artisan queue:work”`

There are several ways and different commands to configure the queue worker in Laravel. For example, specifying maximum job attempts can be done adding “`--tries=3`” to the above command. Official documentation can be found in following URL. <https://laravel.com/docs/5.5/queues>

- Step five – Launching the System

There are two methods of launching the system. Please use of these following methods to launch the system. Please refer Appendix C - User Manual to get the idea about how to operate the system.

## 1. Through the virtual host

- Go to the virtual host creation of your server configurations. For example, in WAMP server go to the following URL in web browser: [http://127.0.0.1/add\\_vhost.php?lang=english](http://127.0.0.1/add_vhost.php?lang=english)
- Then add name of the virtual host, absolute file path and IP address according to the given instructions.
- Now you can directly enter the above given IP address in web browser’s address bar to launch the system.

## 2. Normal method

- Verify the server is running.
- Open the installed web browser and type the URL <http://localhost/ResultsManagementSystem> and press “Enter” button to access the system.

# Appendix B – Design Documentation

## Use case diagrams and description

Followings are the use case diagrams in detail shown in chapter three.

- **Use case diagram for result publishing**

Result publishing is very important and major process of the Results Management System. Examination branch is the authorized person to publish undergraduate's results. Figure B.1 shows the use case of results publishing. The description of the use case is done by Table B.1 and Table B.2.

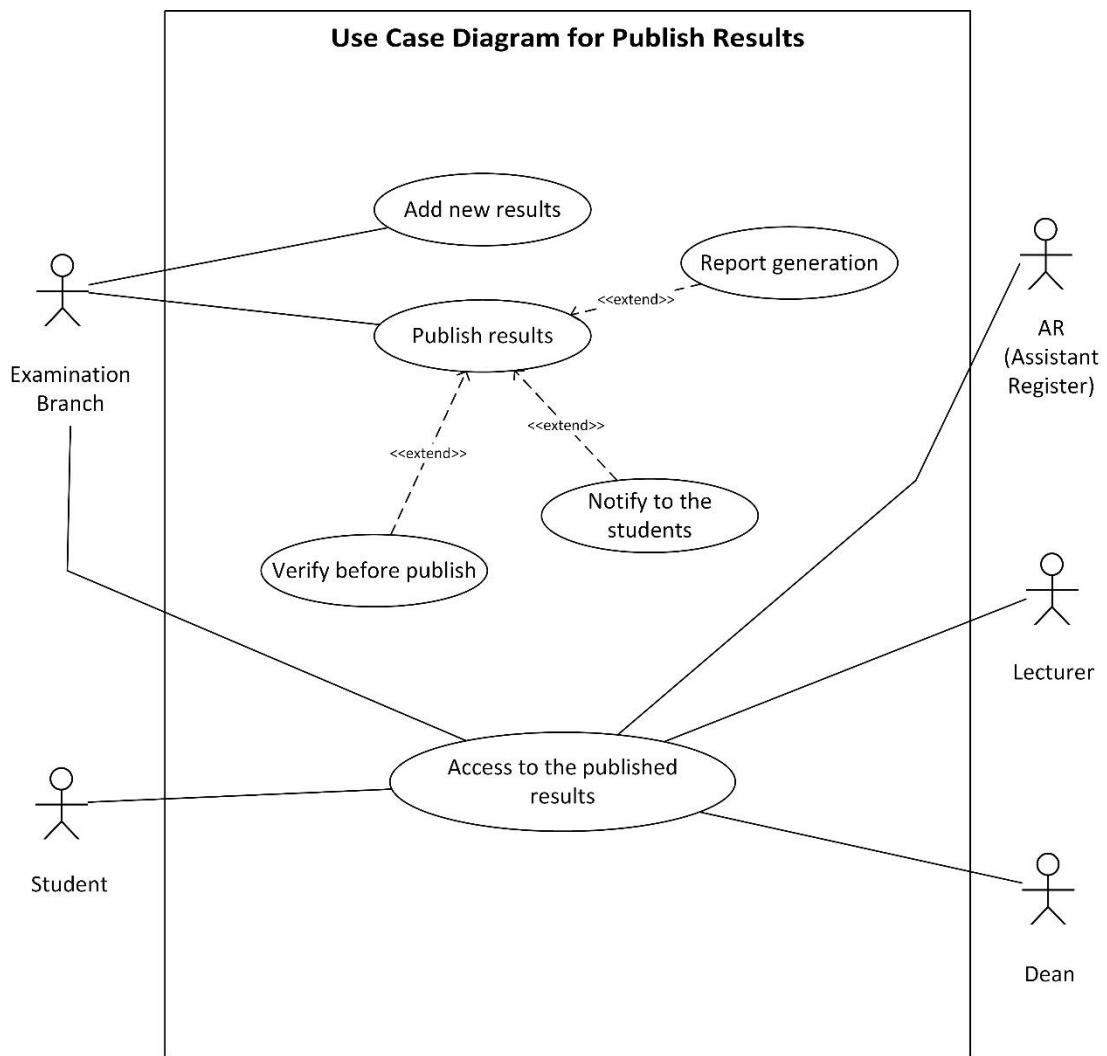


Figure B.1 Use case diagram for result publishing

<b>Use Case</b>	Publishing results
<b>Actor</b>	Examination branch
<b>Description</b>	Add new results to the system and publish
<b>Pre-Conditions</b>	
Check available students and responsible person should verify results of the faculty.	
<b>Flow of Events</b>	
<ol style="list-style-type: none"> <li>1. New results should add according to the exam year.</li> <li>2. If there any special reasons, in the grade field it should be noted. For example, medical as MC, deferred as DFR, absent as AB.</li> <li>3. User can publish results immediately or save and publish later.</li> </ol>	
<b>Post Conditions</b>	
Results should store accurately for the report generation and GPA calculation. Students should get notified new results publish alerts.	

*Table B.1 Use case description - Publishing results*

<b>Use Case</b>	Access to the results
<b>Actor</b>	Examination branch, Dean, AR, Lecturer, Student
<b>Description</b>	Accessing published results of the system
<b>Pre-Conditions</b>	
Results should add and published by the examination branch.	
<b>Flow of Events</b>	
<ol style="list-style-type: none"> <li>1. Examination branch, Dean, AR and Lecturer is authorized to access all published results through the results viewer as well as reports module.</li> <li>2. Student is authorized to view their own results only.</li> </ol>	
<b>Post Conditions</b>	
Authorized persons can generate reports through the reports module.	

*Table B.2 Use case description - Access to the publishing results*

- **Use case diagram for student details management**

Student or in another way undergraduates are the key factor of the university system. Managing their details is very important for the faculty. Following Figure B.2 shows the use case for student details management. There are several actors involving to this process. In the Table B.3 and Table B.4 shows the description of the use case.

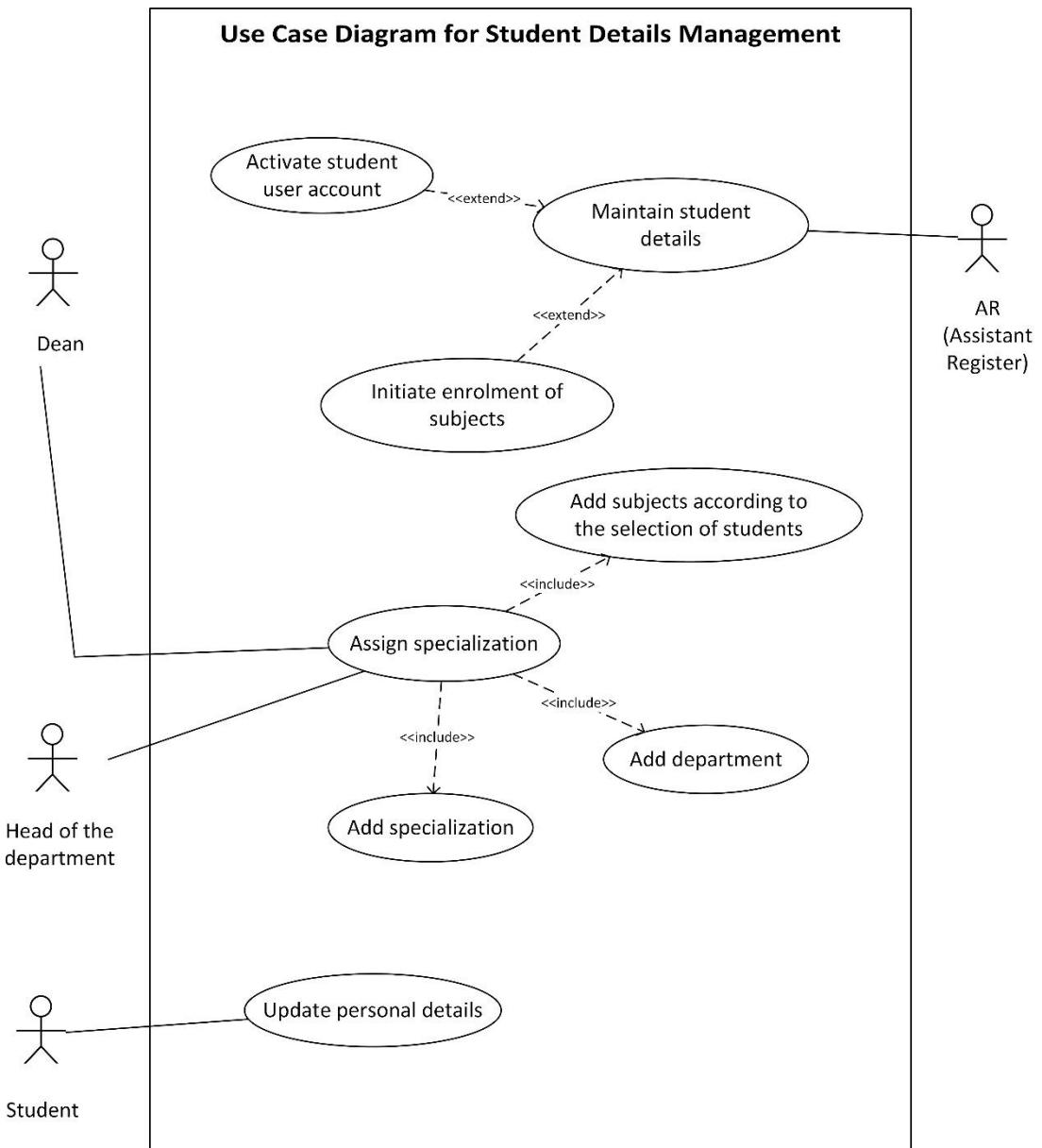


Figure B.2 Use case diagram for managing student details

<b>Use Case</b>	Maintain student details
<b>Actor</b>	AR
<b>Description</b>	Initiate and maintain student details
<b>Pre-Conditions</b>	
Particular student should register by the authorized parties in the university.	
<b>Flow of Events</b>	
<ol style="list-style-type: none"> <li>1. Verify each student details.</li> <li>2. Activate student user account.</li> <li>3. Initiate enrolment of subjects of a student.</li> <li>4. Add student details to the system.</li> </ol>	

5. Maintain student details periodically.
-------------------------------------------

<b>Post Conditions</b>
------------------------

Details should properly save for the future references and report generations.
--------------------------------------------------------------------------------

*Table B.3 Use case description - Maintain student details*

<b>Use Case</b>	Assign specialization
<b>Actor</b>	Dean, Head of the department
<b>Description</b>	Assign specialization to the student
<b>Pre-Conditions</b>	
Student profile should have added to the system.	
<b>Flow of Events</b>	
<ol style="list-style-type: none"> <li>1. Add department to the student based on student preference.</li> <li>2. Add specialization area to the student.</li> <li>3. Enroll and maintain subjects based on the selection of student and other procedures.</li> </ol>	
<b>Post Conditions</b>	
Subject assignment should have done correctly for each student. Because it is the basement of maintain student results.	

*Table B.4 Use case description - Assign specialization*

- **Use case diagram for system configuration**

System configuration should have done by top level user of the system in this case Dean is the responsible person. Configurations are critical to initiate the system. Following Figure B.3 shows the use case diagram for system configurations. Table B.5 describe the use case diagram of system configurations.

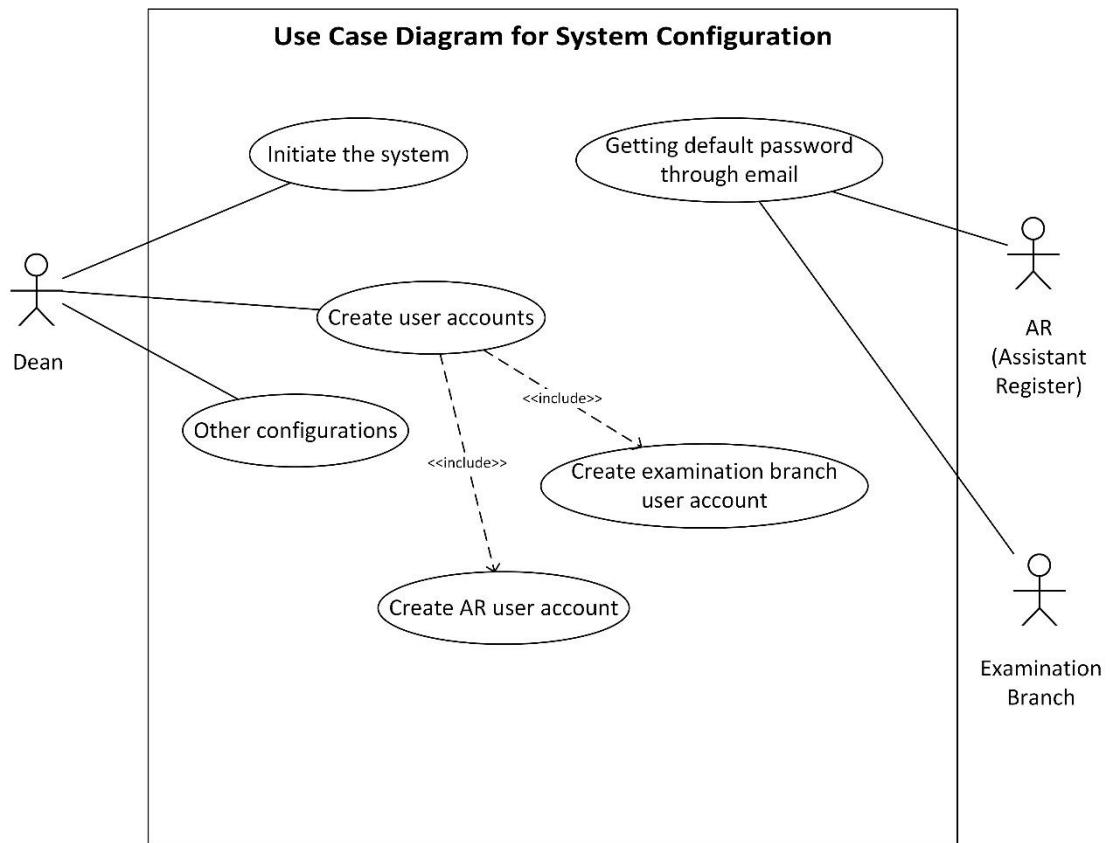


Figure B.3 Use case diagram for system configuration

<b>Use Case</b>	System configuration
<b>Actor</b>	Dean
<b>Description</b>	Initiate system configuration
<b>Pre-Conditions</b>	
System should have up and running correctly.	
<b>Flow of Events</b>	
<ol style="list-style-type: none"> <li>1. Initiate the system by creating Dean's profile.</li> <li>2. Create AR &amp; Examination branch profiles.</li> <li>3. Proceed other configurations. For example, grading system, initiate academic years, courses and its subjects etc.</li> </ol>	
<b>Post Conditions</b>	
Every critical configuration should complete to get better performance of the system.	

Table B.5 Use case description - System configuration

# Appendix C – User Manual

## 1. Configurations

System configuration is very essential process to up and running smooth application. When the initiating system top level user is responsible to complete this process. In this case Dean is the responsible person. Configuration process divided into nine steps for the better user experience. Following steps explains entire configuration process. After the completing all the steps of configuration user will redirect to the dashboard.

### 1.1. Step one - Dean's profile creation

As the first step system needed top level user account to configure rest of the configurations. Following Figure C.1 shows dean's profile creation user interface.

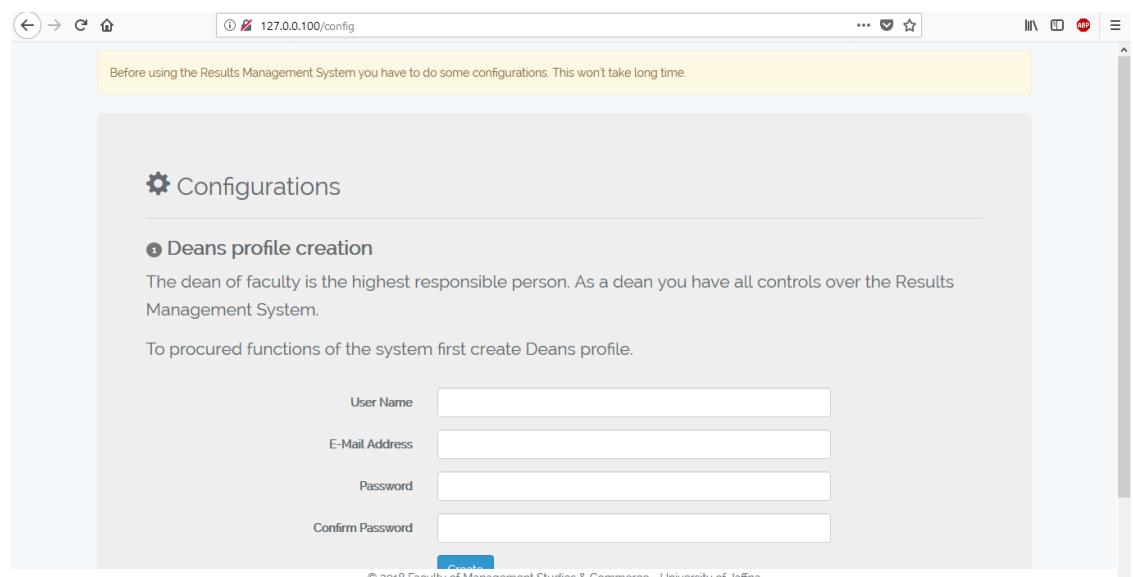


Figure C.1 Dean's profile creation

### 1.2. Step two - Assistant register (AR) profile creation

AR is the person who works on behalf of the Dean to get things work. Dean is responsible person to create AR's profile. Figure C.2 shows AR profile creation window. After the account creation AR get default password from the system through given email. Figure C.3 shows the email received by AR.

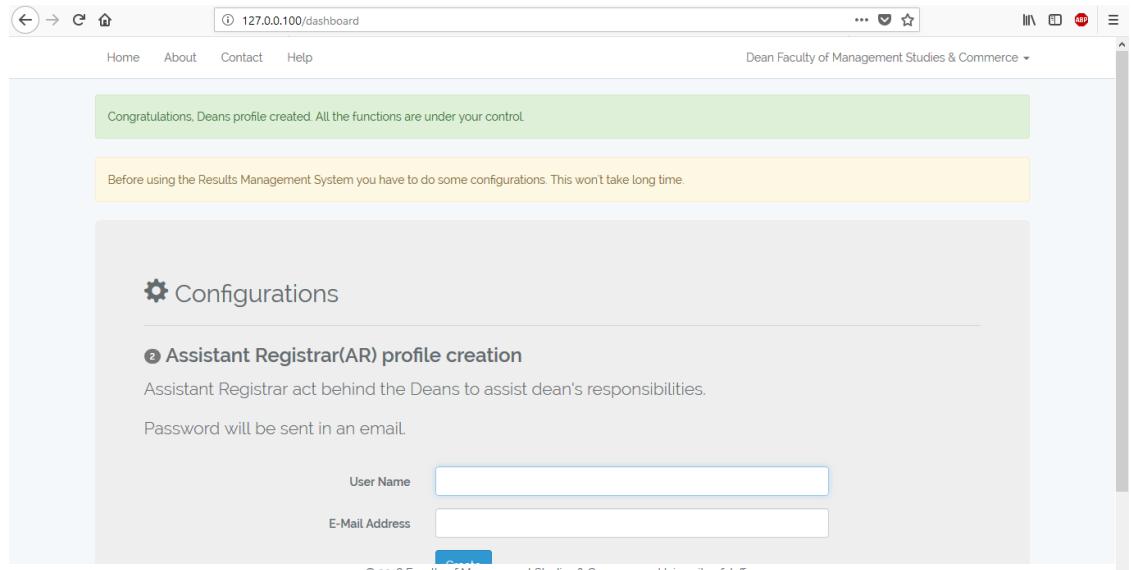


Figure C.2 AR profile creation

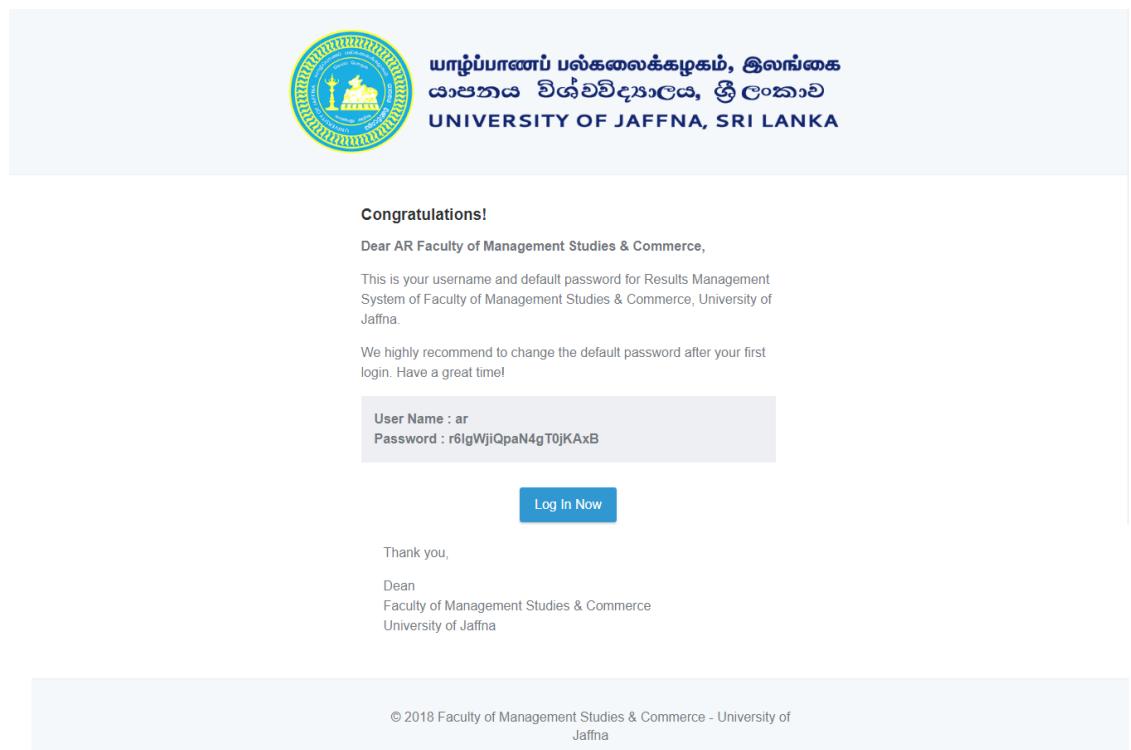


Figure C.3 AR profile creation - Default password received from the system

### 1.3. Step three - Examination branch profile creation

In the step three the Dean has to create profile for Examination Branch. Following Figure C.4 shows the interface for creation of Examination Branch profile. After the creation of profile password will send through given email.

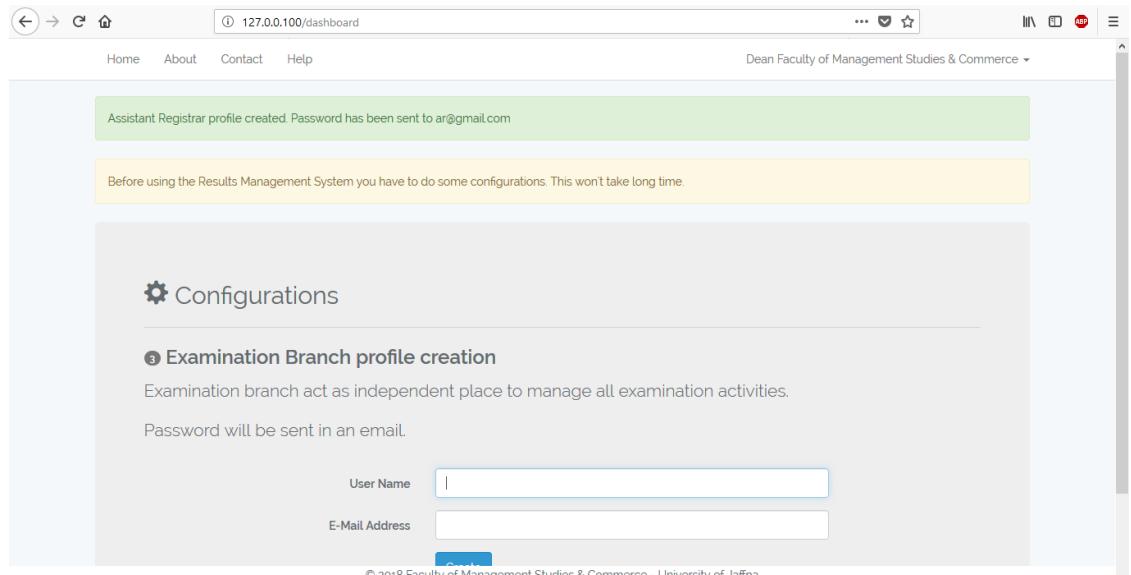


Figure C.4 Examination branch profile creation

#### 1.4. Step four - Course creation

Faculty may offer several courses to their undergraduates. Here the interface to add course details to the system. Figure C.5 shows the courses creation step.

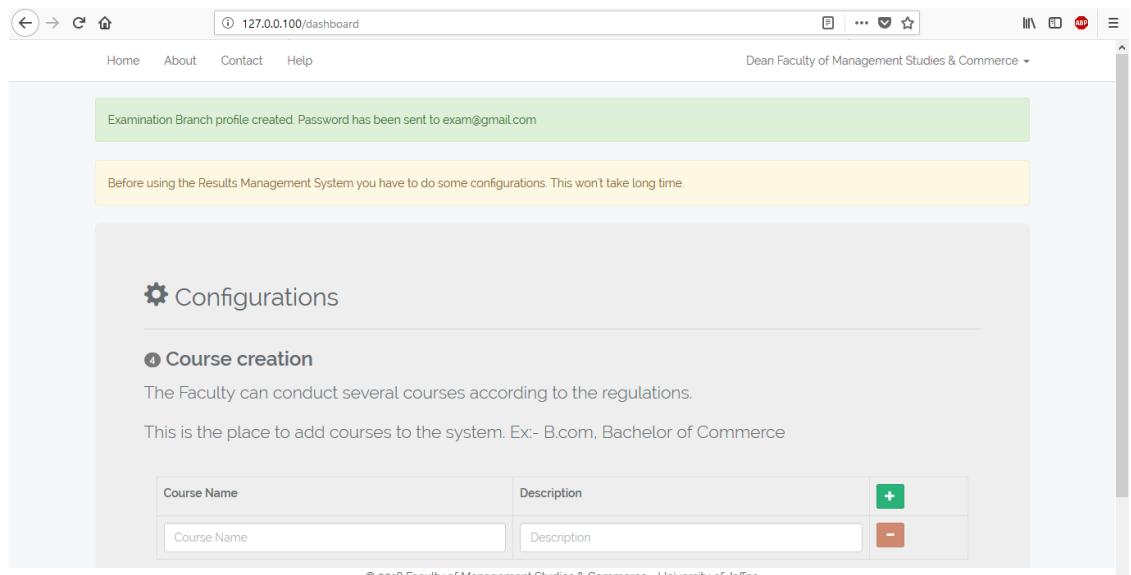


Figure C.5 Course creation

#### 1.5. Step five - Department creation

Faculty belongs to several departments which conduct academic activities in the faculty. Following figure C.6 shows the interface to add department details to the system.

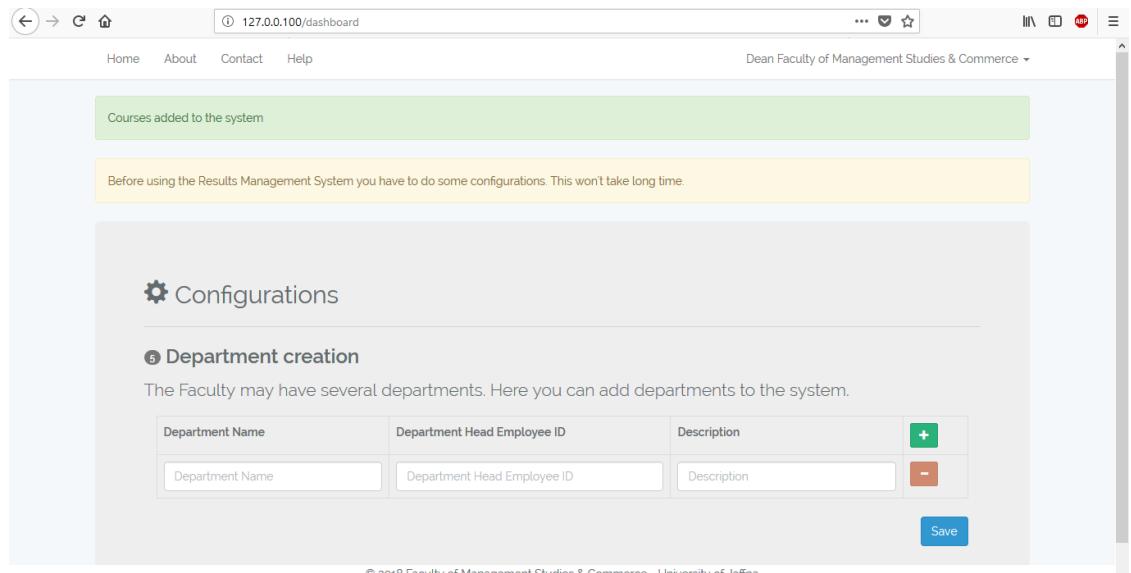


Figure C.6 Department creation

## 1.6. Step six - Specialized area creation

Specialized areas are the sections undergraduates are supposed to work on in their future. Figure C.7 shows the user interface to add specialized area details to the system in configuration state.

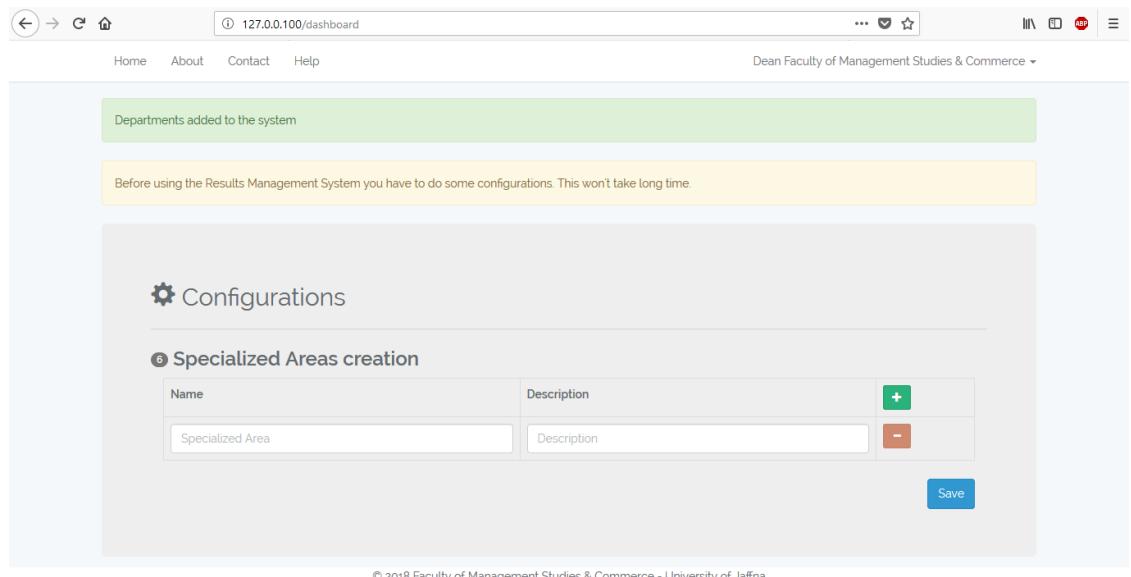


Figure C.7 Specialized areas creation

## 1.7. Step seven - Subject creation

Here is the step to add subject details of each course. Figure C.8 shows the interface to add subject details.

The screenshot shows the 'Configurations' section for 'Subject creation'. A green header bar at the top says 'Specialized Areas added to the system'. Below it is a yellow message bar: 'Before using the Results Management System you have to do some configurations. This won't take long time.' The main area has a heading 'Configs' and a sub-section '7 Subject creation'. It says 'Each an every course has different subjects' and 'This is the place to add subjects to the course.' Below is a table with the following columns: Course, Subject Code, Title, Specialized Area, Conducted By, Credits, Status, Year, Semester, and actions (+, -). The table has several dropdown menus and input fields.

Figure C.8 Subjects creation

### 1.8. Step eight - Grading system creation

There should be proper grading system to evaluate undergraduates' grades. Step nine provide user interface to add grading system to the system. Figure C.9 shows the user interface of grading system creation.

The screenshot shows the 'Configurations' section for 'Grading System creation'. A green header bar at the top says 'Subject added to the system'. Below it is a yellow message bar: 'Before using the Results Management System you have to do some configurations. This won't take long time.' The main area has a heading 'Configs' and a sub-section '6 Grading System creation'. It says 'There should be a well-defined grading system to evaluate student marks.' Below is a table with the following columns: Grade and Points, and actions (+, -). The table has two input fields: 'Grade' and 'Grade Points per Credit'. A blue 'Save' button is at the bottom right.

Figure C.9 Grading system creation

### 1.9. Step nine - Academic years' creation

At the final step user can add initial academic year or years to the system. Every and each student in the faculty are assigned to one academic year. Figure C.10 shows the interface to add academic years to the system.

The screenshot shows a web browser window with the URL `127.0.0.100/dashboard`. The page title is "Dean Faculty of Management Studies & Commerce". At the top, there are links for Home, About, Contact, and Help. A green header bar says "Grades added to the system". Below it, a yellow bar contains the message: "Before using the Results Management System you have to do some configurations. This won't take long time." The main content area has a heading "⚙️ Configurations" and a sub-section "⌚ Academic Years creation". It says: "Before add students to the system you should have at least one academic year. Here you can add academic years." Below this, a note says: "When starting new academic year you can add new academic year to the system from settings." A table is shown for managing academic years:

Year	Description	<a href="#">+</a>
Academic Year	Description	<a href="#">-</a>

At the bottom, a copyright notice reads: "© 2018 Faculty of Management Studies & Commerce - University of Jaffna".

Figure C.10 Academic year creation

## 2. Authentication system

There are several types of users involving to this system namely, Dean, AR, Lecturer, Student, Head of the department, Examination branch. Figure C.11 shows the log in interface of the system. User has to use given login credentials in the configurations. Students in the faculty can log in to the system using their student registration number issued by the faculty as user name and also password. Students are highly advised to change the default password after first log in.

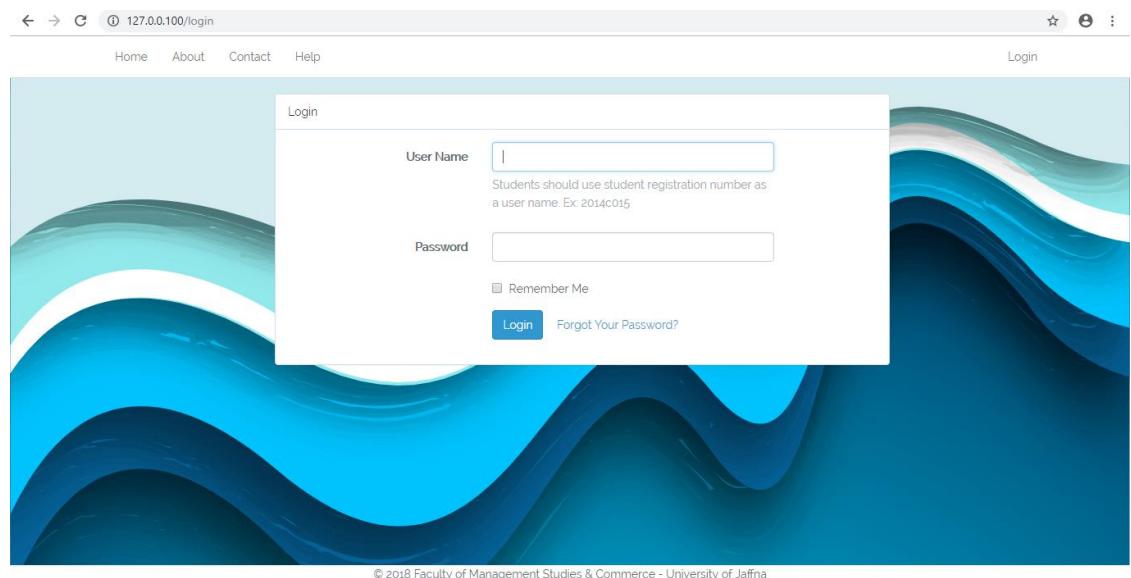
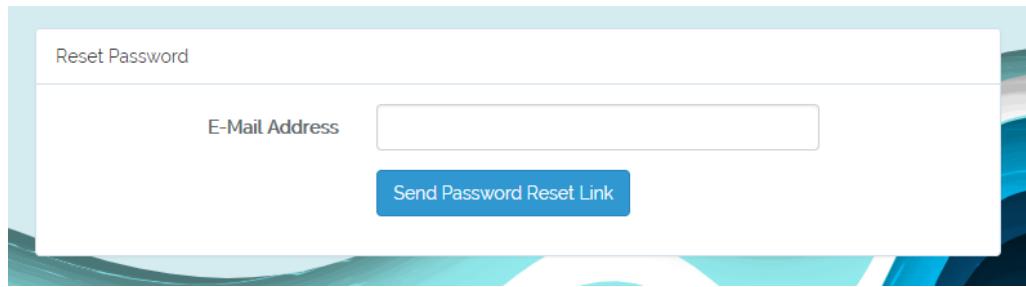


Figure C.11 Log in interface of the system

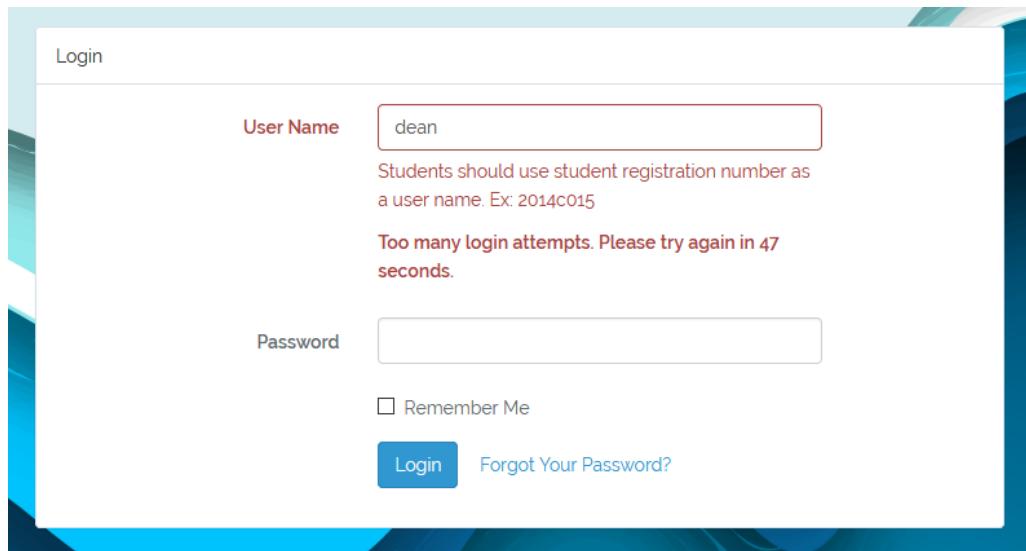
System provide reset password feature to the users when needed to recover their password. Figure C.12 shows the user interface for resetting password. Link to reset password is given in log in interface as "Forgot Your Password?".



A screenshot of a web-based password reset form titled "Reset Password". It features a large input field for "E-Mail Address" and a blue "Send Password Reset Link" button below it.

Figure C.12 Reset log in password

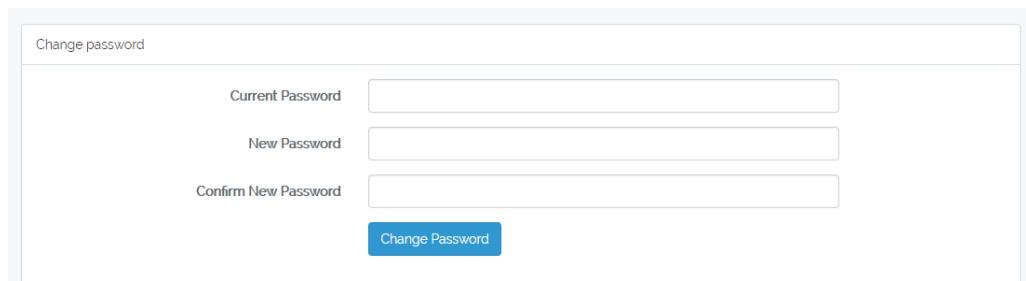
To enhance the security of system it is using three maximum login attempts and lock down the system for one minute. Following Figure C.13 shows the too many login attempts error message.



A screenshot of a login page titled "Login". The "User Name" field contains "dean" and has a red border. A red error message below it says: "Students should use student registration number as a user name. Ex: 2014c015". Another red message below that says: "Too many login attempts. Please try again in 47 seconds." The "Password" field is empty. Below the fields are "Remember Me" and "Login" buttons, and a "Forgot Your Password?" link.

Figure C.13 Too many login attempts error

Following figure C.14 shows password changing feature with can be found in “Profile” section in the dashboard in every user.



A screenshot of a "Change password" form. It includes fields for "Current Password", "New Password", and "Confirm New Password", each with its own input box. A blue "Change Password" button is located at the bottom right.

Figure C.14 Change password

### 3. Navigation pane

Navigation pane provide easy navigation between different pages in the system. This may vary among different user types. Figure C.15 shows navigation pane of user Dean.

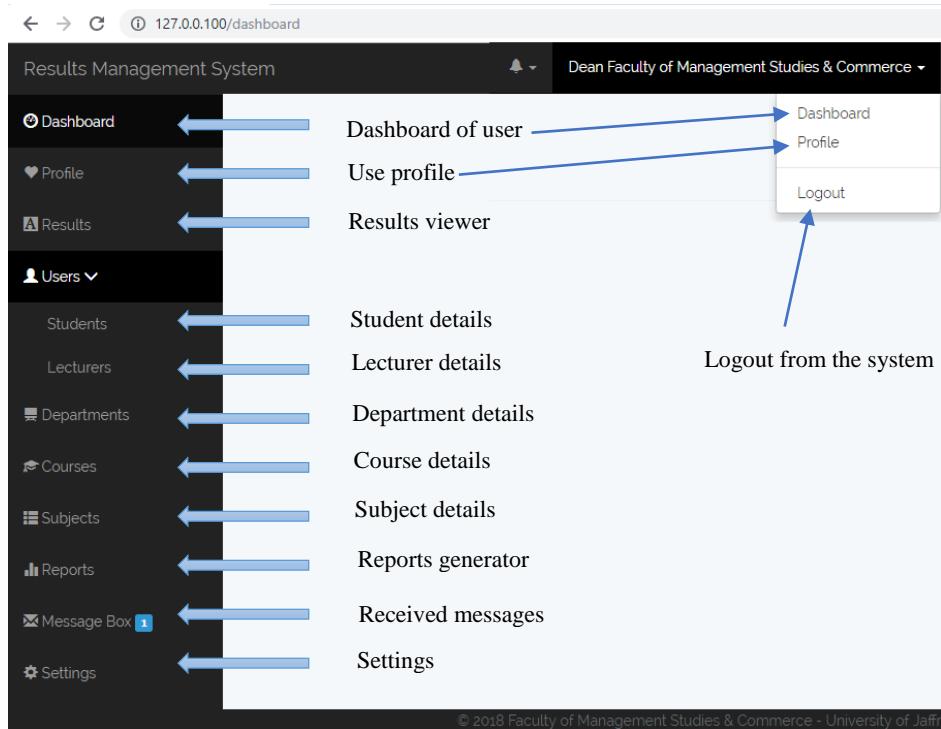


Figure C.15 Navigation pane of the system

### 4. Maintain student details

Maintain student details involve add, update and also delete students from the system. Normally the Dean and AR are authorized to maintain student details while others are authorized to view student details.

#### 4.1. Add student to the system

Users → Students → And New Student → Save ( Figure C.16 shows Add New Student page)

#### 4.2. Edit existing student

Users → Students → Select or Search Student → Edit → Update

#### 4.3. Delete student from the system

Users → Students → Select or Search Student → Delete → Confirm deletion

#### 4.4. View full details of the student

Users → Students → Select or Search Student → View

The screenshot shows the 'Add New Student' form in a web application. The left sidebar contains navigation links for Dashboard, Profile, Results, Users, Departments, Courses, Subjects, Reports, and Message Box. The main content area has a title 'Add New Student' and a breadcrumb trail: Dashboard / Students / Add New. The 'Academic Details' section contains fields for Registration No. (Ex2014C015), Index No. (Exc14015), Department (Not Specified), Course (B.com), Academic Year (2013/2014), and Specialized Area (Not Specified). The 'Personal Details' section contains fields for Initials and First Name. At the bottom, there is a copyright notice: © 2018 Faculty of Management Studies & Commerce - University of Jaffna.

Figure C.16 Add new student to the system

### 5. Maintain lecturer details

Maintain lecturer details involve add, update and also delete lecturer from the system. Normally the Dean and AR are authorized to maintain lecturer details.

#### 5.1. Add new lecturer to the system

Users → Lecturer → Add New Lecturer → Save ( Figure C.17 shows Add New Lecturer page)

#### 5.2. Edit existing lecturer

Users → Lecturer → Select or Search Lecturer → Edit → Update

#### 5.3. Delete lecturer from the system

Users → Lecturer → Select or Search Lecturer → Delete → Confirm deletion

#### 5.4. View full details of the lecturer

Users → Lecturer → Select or Search Lecturer → View

The screenshot shows the 'Results Management System' interface. On the left is a dark sidebar with navigation links: Dashboard, Profile, Results, Users, Departments, Courses, Subjects, Reports, Message Box (with 1 notification), and Settings. The main content area has a title 'Add New Lecturer' and a breadcrumb trail: Dashboard / Lecturers / Add New. Below this is a 'Personal Details' form with fields for Employee ID, Initials, First Name, Last Name, Email, Department, Mobile, and Gender (Male selected). A green success alert box in the top right corner says 'Success Alert' with the message 'Lecture created. Password has been sent to kumanim@gmail.com'. At the bottom of the page is a footer with the text '© 2018 Faculty of Management Studies & Commerce – University of Jaffna'.

Figure C.17 Add new lecturer to the system

## 6. Maintain department details

The faculty having several departments. There are features in the department section to add, edit and delete department details. Figure C.18 shows the user interface of department section.

### 6.1. Add new department to the system

Departments → And New Departments → Save

### 6.2. Edit existing department

Departments → Edit → Update

### 6.3. Delete department from the system

Departments → Delete → Confirm deletion

Department Name	Department Head Employee ID	Description	Actions	
Commerce	EMP001	Department of Commerce	Edit	Delete
Accounting	EMP002	Department of Accounting	Edit	Delete
HR	EMP003	Department of Human Resource	Edit	Delete
Marketing	EMP004	Department of Marketing	Edit	Delete
Finance	EMP005	Department of Finance	Edit	Delete

© 2018 Faculty of Management Studies & Commerce - University of Jaffna

Figure C.18 Department details

## 7. Maintain courses details

The faculty offering several courses to their undergraduates. Here you can maintain courses details.

7.1. Add new course to the system (Figure C.19 shows user interface for adding new course to the system)

Courses → Add New Course → Save

7.2. Edit existing course

Courses → Edit → Update

7.3. Delete course from the system

Courses → Delete → Confirm deletion

The screenshot shows a web-based application titled "Results Management System" for "Dean Faculty of Management Studies & Commerce". The left sidebar contains navigation links: Dashboard, Profile, Results, Users, Departments, Courses (which is selected), Subjects, Reports, Message Box (with a notification count of 1), and Settings. The main content area is titled "Add New Course" and shows a breadcrumb trail: Dashboard / Courses / Add New. Below this is a table with two columns: "Course Name" and "Description". Each column has an input field and a row of buttons for adding (+) or removing (-) items. At the bottom right of the table is a "Save" button.

*Figure C.19 Add new course to the system*

## 8. Maintain subject details

The faculty offering several courses to their undergraduates and each course having different subjects according to semester. Here subjects can have added according to the course and semester. Figure C.20 shows subject details according to the added courses.

### 8.1. Add new subject to the system

Subjects → And New Subject → Save

### 8.2. Edit existing subject

Subjects → Edit → Update

### 8.3. Delete subject from the system

Subjects → Delete → Confirm deletion

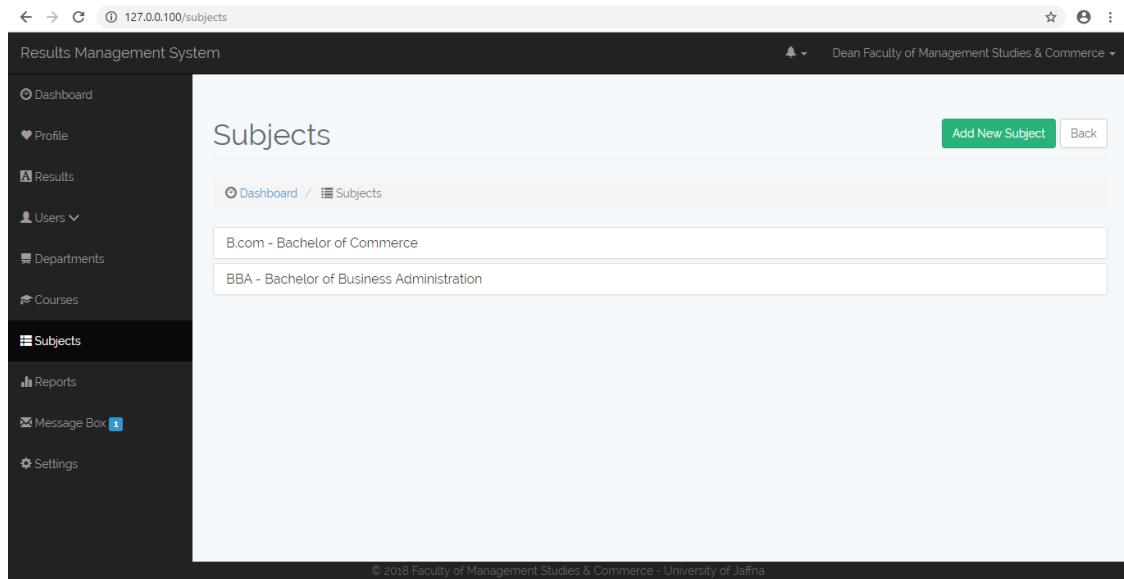


Figure C.20 Subject details

## 9. Maintain student results

Examination branch is the responsible parties to maintain student results. There are several operations involving in managing results.

### 9.1. Add student results

Adding student result is done by finding list of students according to the academic year. Then after it will redirect to the student name list with the field to add grade. Examination year should mention in top of the page.

Figure C.21 shows the link to find relevant students for the purpose of adding grades. The redirected page from the Figure C.21 shows in Figure C.22. There are two options in the adding results. First one is only saving results and it will appear in “Publish results” area. Second one is save and publish immediate those results. This will send notify relevant student regarding new subject results through system notifications and emails. Figure C.23 shows notification getting by authorized party after successfully saving results.

Figure C.21 Add new results

Results Management System

Students - Add Results

Subject: BBAA 2243 - Management Accounting  
Academic Year: 2013/2014

Registration No.	Index No.	Academic Year	First Name	Last Name	Course	Department	Grade
2014bad001	bads14001	2013/2014	Kumara	Silva	BBA	Not Specified	A
2014bad035	bads14035	2013/2014	Tharindu	Kaushalya	BBA	Not Specified	B+

Showing 1 to 2 of 2 entries

Exam Year: 2018

Search: [ ]

Previous [1] Next

Save & Publish Save

Figure C.22 Add new results - Enter new grades

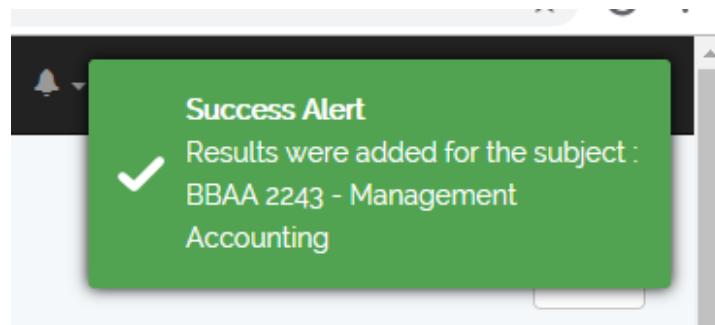


Figure C.23 Notification - Results added message

## 9.2. Edit student results

Edit student results can be done through the figure showing in Figure C.24 by searching results according to the exam year.

Edit Results

Select Subject: BBAA 1133 - Financial Accounting i

Select Exam Year: 2019

View list of students with results according to subject and exam year.

View Results

Figure C.24 Edit student results

## 9.3. Add repeating subject results

Adding repeating subject result can be done in separate window. Because all the students are not repeating subjects and it is not possible to get student list

according to academic year without entering grades for the security reasons. Figure C.25 show the interface for entering grades for repeating subjects.

Add Repeat Subject Results

**Student Registration No. \*** Enter Student Registration Number

**Exam Year \*** Enter Exam Year

**Select Subject \*** BBAA 1133 - Financial Accounting i

**Select Grade \*** MC

This should use to add repeat subjects only.

**Save & Publish** **Save**

Figure C.25 Add repeating subject results

#### 9.4. Publish results

Publishing result may set results are accessible state and notify relevant student about new results. Figure C.26 shows publish results window.

Results Management System

Do you really want to publish?

Publish results of BBAA 2243-Management Accounting in exam year 2018

**Publish Results** BBAA 2243-Management Accounting in exam year 2018

**Add New Results**

Select Subject \* BBAA 1133 - Financial Accounting i

Select Academic Year \* 2013/2014

Show list of students according to subject and academic year Show Students

**Add Repeat Subject Results**

**Student Registration No. \*** Enter Student Registration Number

**Exam Year \*** Enter Exam Year

Figure C.26 Publish results

#### 9.5. Search student results

Search student results can have done in several ways. They are, With student registration number, student index number, student name and subject. Figure C.27 shows search options of student results. Through “View” button in the search results window users can see full results of selected student. Figure C.28 and Figure C.29 shows full results of a student.

Search Options

Search By: Student Registration Num.

Registration No.:

Figure C.27 Search options - Student Results

127.0.0.100/grades/2014c015

Results Management System

Dashboard Results Profile GPA

## View Results

[Download](#)

Student Name: S.M. Madumal Jeewantha  
 Registration Number: 2014c015  
 Index Number: c14015  
 Academic Year: 2013/2014  
 Course: B.com  
 Specialized Area: BTM  
 Department: Commerce



Dashboard / Results / 2014c015

© 2018 Faculty of Management Studies & Commerce – University of Jaffna

Figure C.28 Full results view - Part one

127.0.0.100/grades/2014c015

Results Management System

Dashboard Results Profile GPA

Dashboard / Results / 2014c015

Year 1 - Semester 1				
Subject Code	Title	Credits	Exam Year	Grade
COM 11031	Business Environment	3	2015	A
COM 11032	Economic for Enterprise	3	2015	B
COM 11033	Principles and Practices of Banking	3	2015	B
COM 11034	Business Mathematics and Statistics	3	2015	B-
COM 11035	Financial Accounting - i	3	2015	C
COM 11026	Business Communication - i	2	2015	C

Year 1 - Semester 2				
© 2018 Faculty of Management Studies & Commerce – University of Jaffna				

GPA

- First Class : 3.70
- Second Class(Upper Division) : 3.30
- Second Class(Lower Division) : 3.00
- Current State : 3.2576470588235

Attention

Figure C.29 Full results view - Part two

## 10. Dashboard

According to the user types dashboards may provide different contents. Dashboards provide easy access to the essential functionalities of users. Following figure C.30 shows Dean's dashboard and Figure C.31 shows student dashboard.

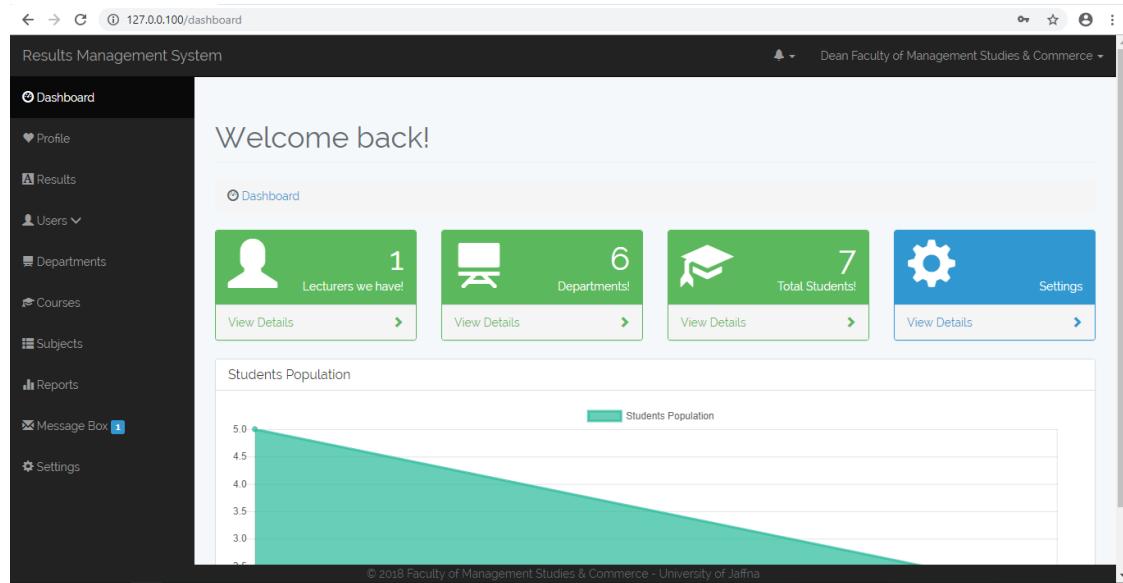


Figure C.30 Dashboard - Dean

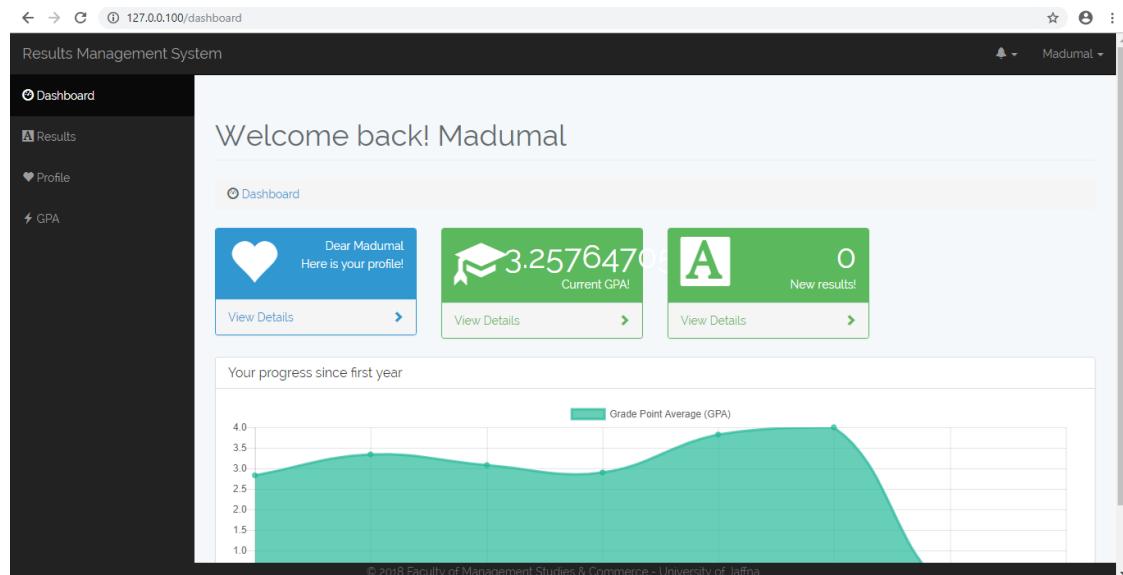


Figure C.31 Dashboard - Student

## 11. Settings

There are three types of settings. General settings are accessible to the Dean only and Head of the Department also has permission to access rest of the settings. Figure C.32 shows index page of settings.

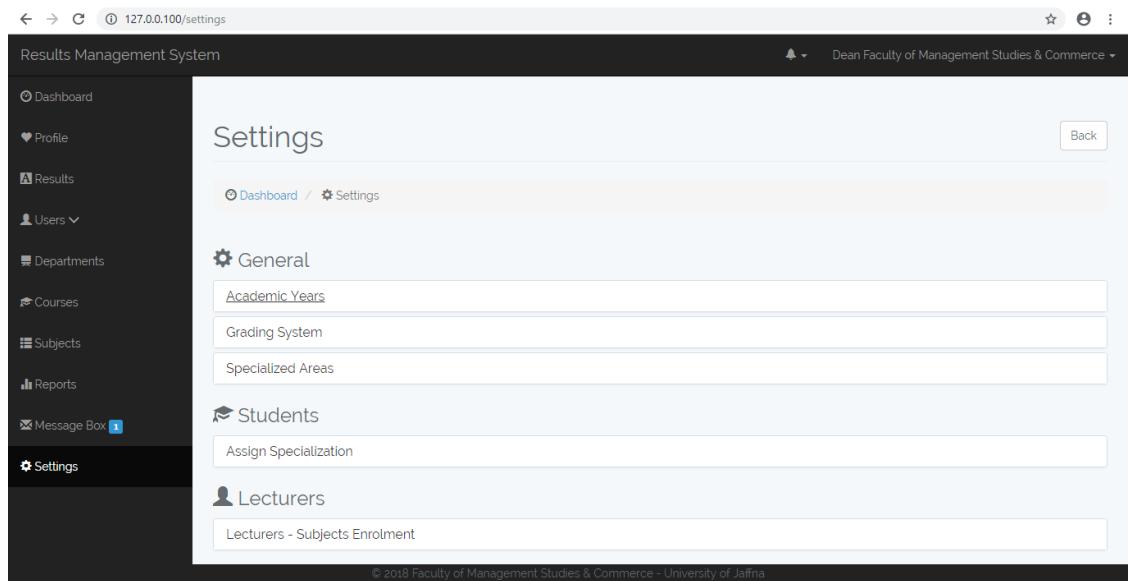


Figure C.32 Settings page

### 11.1. General settings

In the general setting there are Academic years, Grading system and Specialized areas settings.

### 11.2. Student settings

Here you can do specializations for students. It will include setup department, specialization area and its subjects. This process should have done by Dean or Head of the Department only. This may use Academic Year and Course to filter out students.

### 11.3. Lectures settings

Here you can do enrolment for lecturers. A lecturer can conduct one or more subjects and this may vary among academic years. This process should have done by Dean or Head of the Department only.

## 12. Reports

Reports are categories as following titles. To the generation of reports relevant columns can be select prior to the generation. Reports can extract in PDF and Excel form. Figure C.33 shows the selection of columns to generate report students in specific course.

- Student Details
- Lecturers Details
- Results
- Course Details
- Grading System

The screenshot shows a web-based application titled "Results Management System". The left sidebar contains navigation links: Dashboard, Profile, Results, Users, Departments, Courses, Subjects, Reports (selected), Message Box (1), and Settings. The main content area is titled "Students in specific course :". It includes filters for "Select Academic Year" (2013/2014) and "Course" (B.com). Below these are two columns of checkboxes for selecting columns:

Selected Columns	Available Columns
<input checked="" type="checkbox"/> Registration Number	<input type="checkbox"/> Date of Birth
<input checked="" type="checkbox"/> Index Number	<input type="checkbox"/> Gender
<input checked="" type="checkbox"/> Department	<input type="checkbox"/> Marriage State
<input checked="" type="checkbox"/> Course	<input type="checkbox"/> Email
<input checked="" type="checkbox"/> Academic Year	<input type="checkbox"/> Contact No Mobile
<input checked="" type="checkbox"/> Specialized Area	<input type="checkbox"/> Contact No Home
<input checked="" type="checkbox"/> Initials	<input type="checkbox"/> Home Address 1
<input checked="" type="checkbox"/> First Name	<input type="checkbox"/> Home Address 2
<input checked="" type="checkbox"/> Last Name	<input type="checkbox"/> Home Address 3
<input checked="" type="checkbox"/> NIC Number	<input type="checkbox"/> Current Address 1
	<input type="checkbox"/> Current Address 2
	<input type="checkbox"/> Current Address 3
	<input type="checkbox"/> FB Profile
	<input type="checkbox"/> LinkedIn Profile
	<input type="checkbox"/> Father Name
	<input type="checkbox"/> Father Occupation
	<input type="checkbox"/> Mother Name
	<input type="checkbox"/> Mother Occupation
	<input type="checkbox"/> Number of Sister and Brothers
	<input type="checkbox"/> Dissertation Title
	<input type="checkbox"/> Dissertation Published Link
	<input type="checkbox"/> Supervisor Name
	<input type="checkbox"/> Profile Picture
	<input type="checkbox"/> Bio

At the bottom of the page, a footer bar displays the text "© 2018 Faculty of Management Studies & Commerce - University of Jaffna".

Figure C.33 Reports - Selection of columns

# Appendix D – Management Reports

## 1. Student Details

### 1.1. Students in specific academic year

This report will show students in given academic year. Figure D.1 shows students in academic year 2014/2015.

Student Details									
Academic Year : 2013/2014									
Registration No.	Index No.	Department	Course	Academic Year	Specialized Area	Initials	First Name	Last Name	NIC No.
2014bad001	bad14001	Not Specified	BBA	2013/2014	Not Specified	R.T.	Kumara	Silva	947412744V
2014bad035	bad14035	Not Specified	BBA	2013/2014	Finance	D.	Tharindu	Kaushalya	922403767V
2014c001	c14001	Commerce	B.com	2013/2014	BTM	R.T.	Dasuni	Perera	947410852V
2014c002	c14002	Commerce	B.com	2013/2014	Accounting and Finance	R.U.	Rusuru	Rajapkshe	944960288V
2014c015	c14015	Commerce	B.com	2013/2014	BTM	S.M.	Madumal	Jeewantha	942340400V

Figure D.1 Reports - Students in specific academic year

### 1.2. Students in specific course

Through this option user can create report of students in given course. Figure D.2 show students in “B.com” course.

Student Details									
Academic Year : 2013/2014									
Course : B.com									
Registration No.	Index No.	Department	Course	Academic Year	Specialized Area	Initials	First Name	Last Name	NIC No.
2014c001	c14001	Commerce	B.com	2013/2014	BTM	R.T.	Dasuni	Perera	947410852V
2014c002	c14002	Commerce	B.com	2013/2014	Accounting and Finance	R.U.	Rusuru	Rajapkshe	944960288V
2014c015	c14015	Commerce	B.com	2013/2014	BTM	S.M.	Madumal	Jeewantha	942340400V

Figure D.2 Reports - Students in specific course

### 1.3. Students in specific department

Here user can give department and academic year to generate this report. Figure D.3 shows students of “Commerce” department in academic year “2014/2015”.

Student Details									
Academic Year : 2013/2014									
Department : Commerce									
Registration No.	Index No.	Department	Course	Academic Year	Specialized Area	Initials	First Name	Last Name	NIC No.
2014c001	c14001	Commerce	B.com	2013/2014	BTM	R.T.	Dasuni	Perera	947410852V
2014c002	c14002	Commerce	B.com	2013/2014	Accounting and Finance	R.U.	Rusuru	Rajapkshe	944960288V
2014c015	c14015	Commerce	B.com	2013/2014	BTM	S.M.	Madumal	Jeewantha	942340400V

Figure D.3 Reports - Students in specific department

#### 1.4. Students in specific specialized area

There are several specialized areas in the faculty here users can filter out students according to their specialized areas. Figure D.4 shows students of specialized area “BTM” in academic year “2014/2015”.

Student Details									
Academic Year : 2013/2014									
Specialized Area : BTM									
Registration No.	Index No.	Department	Course	Academic Year	Specialized Area	Initials	First Name	Last Name	NIC No.
2014c001	c14001	Commerce	B.com	2013/2014	BTM	R.T.	Dasuni	Perera	947410852V
2014c015	c14015	Commerce	B.com	2013/2014	BTM	S.M.	Madumal	Jeewantha	942340400V

Figure D.4 Reports - Students in specific specialized area

#### 1.5. Full details of a student

Full details of a student can generate with the given registration number. Figure D.5 shows details of student registration number “2014/c/015”.

Student Details									
Student Registration Number: 2014c015									
Registration No.	Index No.	Department	Course	Academic Year	Specialized Area	Initials	First Name	Last Name	NIC No.
2014c015	c14015	Commerce	B.com	2013/2014	BTM	S.M.	Madumal	Jeewantha	942340400V

Figure D.5 Reports - Full details of a student

#### 1.6. Assigned subjects of a student

Students are assigned with different subjects of a course. Here with the registration number can generate list of assigned subjects. Figure D.6 shows the assigned subjects of student registration number “2014/c/015”.

## Student Details - Assigned Subjects

Student Registration Number: **2014c015**

Name: **S.M. Madumal Jeewantha**

Course: **B.com**

Academic Year: **2013/2014**

Specialized Area: **BTM**

Department: **Commerce**

Year 1 - Semester 1

Subject Code	Title	Credits	Status	Conducted By	Status
COM 11026	Business Communication - i	2	Compulsory	Commerce Department	Enrolled

*Figure D.6 Reports - Assigned subjects of a student*

## 2. Lecturers Details

### 2.1. All the lecturers in the faculty

Here can generate report of all the lecturers in the faculty for selected columns. Figure D.7 shows report of all the lectures in the faculty.

#### Lecturers Details

Employee ID	Department.	Initials	First Name	Last Name	Email	Mobile	Gender
EMP001	Commerce	B.	Prahalathan		parahaladan@gmail.com		Male
EMP002	Accounting	R.T.	Nimaladasan	Kumar	kumaniml@gmail.com		Male

*Figure D.7 Reports - All the lecturers in the faculty*

### 2.2. Lecturers in specific department

This report can generate for the lectures in specific department. Figure D.8 shows lectures in the “Commerce” department.

#### Lecturers Details

Department: **Commerce**

Employee ID	Department.	Initials	First Name	Last Name	Email	Mobile	Gender
EMP001	Commerce	B.	Prahalathan		parahaladan@gmail.com		Male

*Figure D.8 Reports - Lecturers in specific department*

### 2.3. Assigned subjects of specific lecturer

Lecturers in the faculty are enrolled with different subjects among different academic years. Following figure D.9 shows assigned subjects of lecturer employee number “EMP001” in academic year “2014/2015”.

Lecturer Details - Assigned Subjects				
Employee ID: <b>EMP001</b>				
Name: <b>B. Prahala</b>				
Department: <b>Commerce</b>				
Mobile:				
Email: <a href="mailto:parahaladan@gmail.com">parahaladan@gmail.com</a>				
Subject Code	Title	Credits	Status	Conducted By
COM ACF 31043	Advanced Corporate Finance	3	Major	Commerce Department

Figure D.9 reports - Assigned subjects of a lecturer

### 3. Results

#### 3.1. Full results of a student

This will give full results sheet of the given student. Figure D.10 shows the full results view of student registration number “2014/c/015”.

Results Sheet				
Student Name: S.M. Madumal Jeewantha				
Registration Number: 2014c015				
Index Number: c14015				
Academic Year: 2013/2014				
Course: B.com				
Specialized Area: BTM				
Department: Commerce				
Current State of GPA : 3.2576470588235				
Year 1 - Semester 1				
Subject Code	Title	Credits	Exam Year	Grade

Figure D.10 Reports - Full results of a student

### 3.2. Subject results

Subject results show all the students assigned to particular subject in given exam year. Figure D.11 shows results of subject “COM 11026 – Business communication I” in exam year “2015”.

Subject Results			
Subject Code : <b>COM 11026</b>	Index No.	Name	Grade
2014c001	c14001	R.T. Dasuni Perera	C+
2014c015	c14015	S.M. Madumal Jeewantha	C

Figure D.11 Reports - Subject results

### 3.3. Semester results

This is very important report in results section. This may provide all the results of semester subjects. Figure D.12 shows results of Year “1”, Semester “1” of “B.com” course in “2015” exam year.

Semester Results									
Course : <b>B.com</b>	Index No.	Name	COM 11026	COM 11031	COM 11032	COM 11033	COM 11034	COM 11035	
2014c001	c14001	R.T. Dasuni Perera	C+	B+	C+	B-	A-	C+	
2014c015	c14015	S.M. Madumal Jeewantha	C	A	B	B	B-	C	

Figure D.12 Reports - Semester Results

## 4. Course Details

This will provide subjects details of given course. Figure D.13 shows details of “B.com” course.

## Course Details

Course: B.com

Year 1 - Semester 1

Subject Code	Title	Credits	Status	Conducted By
COM 11026	Business Communication - i	2	Compulsory	Commerce Department
COM 11031	Business Environment	3	Compulsory	Commerce Department
COM 11032	Economic for Enterprise	3	Compulsory	Commerce Department
COM 11033	Principles and Practices of	3	Compulsory	Commerce

Figure D.13 Reports - Course details

## 5. Grading System

This report provides grading system of the faculty. Figure D.14 shows grading system details.

## Grading System

Grade	Points
MC	0
DFR	0
AB	0
A+	4
A	4
A-	3.7
B+	3.3
R	3

Figure D.14 Reports - Grading system

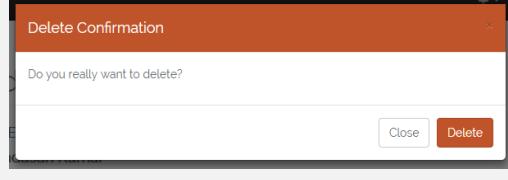
# Appendix E – Test Results

- Test Cases for Configuration Module

No	Test Case	Expected Output	Actual Output	Status
1	Dean's profile creation	Successfully created and send default password.	Congratulations. Deans profile created. All the functions are under your control	Pass
2	AR's profile creation	Successfully created and send default password.	Assistant Registrar profile created. Password has been sent to ar@gmail.com	Pass
3	Exam branch profile creation	Successfully created and send default password.	Examination Branch profile created. Password has been sent to exam@gmail.com	Pass

Table E.1 Test Cases for Configuration Module

- Test Cases for Settings Module

No	Test Case	Expected Output	Actual Output	Status																			
1	Identify specialization assignment	Assigned students showing as red. Not assigned students showing as green.	<table border="1"> <thead> <tr> <th>Course</th> <th>Department</th> <th>Specialization</th> <th>Actions</th> </tr> </thead> <tbody> <tr> <td>com</td> <td>Commerce</td> <td>BTM</td> <td><button>View</button></td> </tr> <tr> <td>com</td> <td>Commerce</td> <td>Not Specified</td> <td><button>View</button></td> </tr> </tbody> </table>	Course	Department	Specialization	Actions	com	Commerce	BTM	<button>View</button>	com	Commerce	Not Specified	<button>View</button>	Pass							
Course	Department	Specialization	Actions																				
com	Commerce	BTM	<button>View</button>																				
com	Commerce	Not Specified	<button>View</button>																				
2	Showing correct students	Students should show according to the given course.	<table border="1"> <thead> <tr> <th>Course</th> </tr> </thead> <tbody> <tr> <td>B.com</td> </tr> <tr> <td>B.com</td> </tr> </tbody> </table>	Course	B.com	B.com	Pass																
Course																							
B.com																							
B.com																							
3	Save assigned subjects	When checking after update there should be correct results.	<table border="1"> <thead> <tr> <th>edits</th> <th>Status</th> <th>Conducted By</th> <th>Select</th> </tr> </thead> <tbody> <tr> <td>Major</td> <td>Commerce Department</td> <td><input checked="" type="checkbox"/></td> </tr> <tr> <td>Major</td> <td>Commerce Department</td> <td><input checked="" type="checkbox"/></td> </tr> <tr> <td>Major</td> <td>Commerce Department</td> <td><input checked="" type="checkbox"/></td> </tr> <tr> <td>Electives</td> <td>Commerce Department</td> <td><input type="checkbox"/></td> </tr> <tr> <td>Electives</td> <td>Commerce Department</td> <td><input checked="" type="checkbox"/></td> </tr> </tbody> </table>	edits	Status	Conducted By	Select	Major	Commerce Department	<input checked="" type="checkbox"/>	Major	Commerce Department	<input checked="" type="checkbox"/>	Major	Commerce Department	<input checked="" type="checkbox"/>	Electives	Commerce Department	<input type="checkbox"/>	Electives	Commerce Department	<input checked="" type="checkbox"/>	Pass
edits	Status	Conducted By	Select																				
Major	Commerce Department	<input checked="" type="checkbox"/>																					
Major	Commerce Department	<input checked="" type="checkbox"/>																					
Major	Commerce Department	<input checked="" type="checkbox"/>																					
Electives	Commerce Department	<input type="checkbox"/>																					
Electives	Commerce Department	<input checked="" type="checkbox"/>																					
4	Enroll lecturer with new subject	Get success notification.		Pass																			
5	Delete enrolled subject from lecture	Get confirmation dialog and success notification.		Pass																			

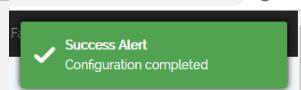
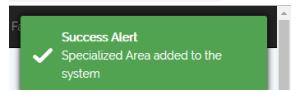
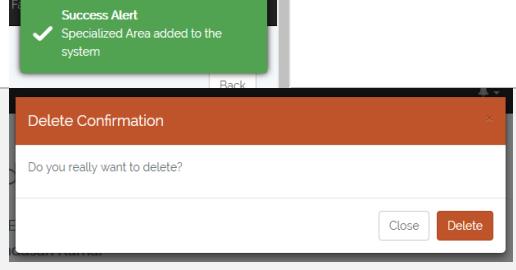
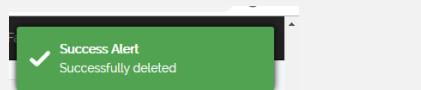
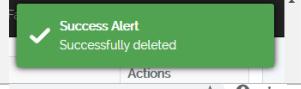
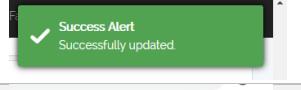
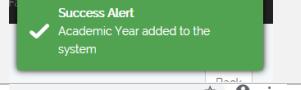
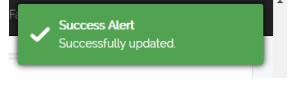
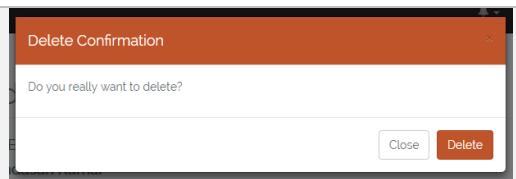
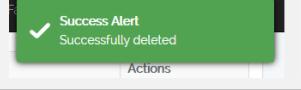
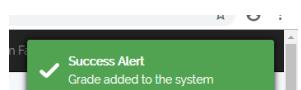
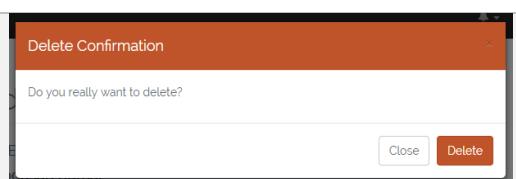
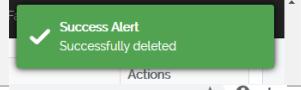
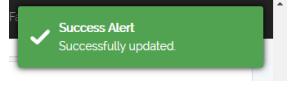
				
6	Add new specialization	Get success notification.	 	Pass
7	Delete specialization	Get confirmation window and success notification.	 	Pass
8	Update existing specialization	Get success notification.		Pass
9	Add new academic year	Get success notification.		Pass
10	Update existing academic year	Get success notification.		Pass
11	Delete academic year	Get confirmation window and success notification.	 	Pass
12	Add new grade in grading system	Get success notification.		Pass
13	Delete existing grade in grading system	Get confirmation window and success notification.	 	Pass
14	Update existing grade in grading system	Get success notification.		Pass

Table E.2 Test Cases for Settings Module

- Test Cases for Notification Module

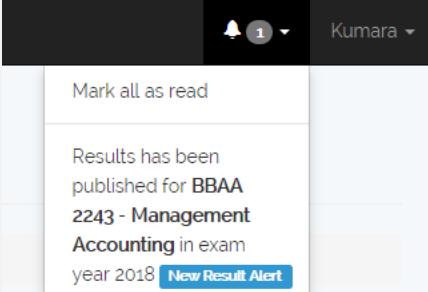
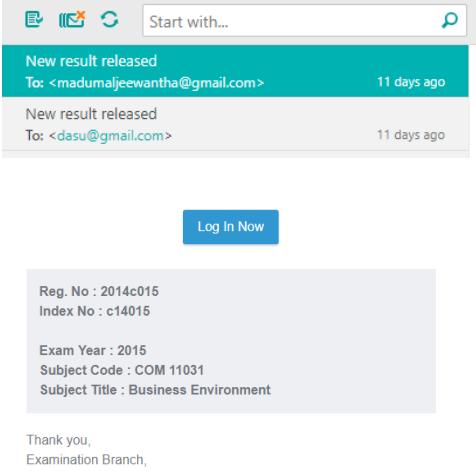
No	Test Case	Expected Output	Actual Output	Status
1	System notification of new results publish	Received notification in student dashboard and show notification count.		Pass
2	Email notification of new results publish	Received email from the system to relevant students.		Pass

Table E.3 Test Cases for Notification Module

- Test Cases for Courses Module

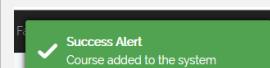
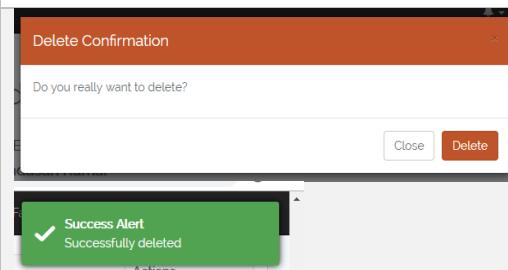
No	Test Case	Expected Output	Actual Output	Status
1	Add new course	Get success notification.		Pass
2	Edit existing course details	Get success notification.		Pass
3	Delete existing course	Get confirmation window and success notification.		Pass

Table E.4 Test Cases for Courses Module

- Test Cases for Departments Module

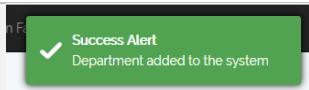
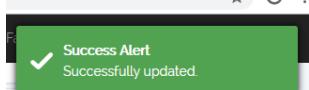
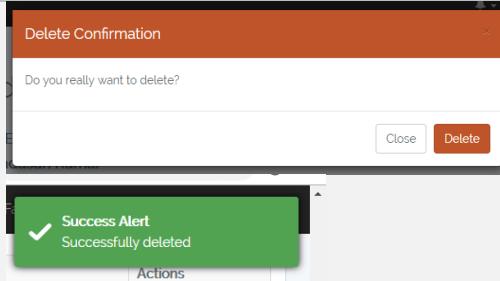
No	Test Case	Expected Output	Actual Output	Status
1	Add new course	Get success notification.		Pass
2	Edit existing course details	Get success notification.		Pass
3	Delete existing course	Get confirmation window and success notification.		Pass

Table E.5 Test Cases for Departments Module

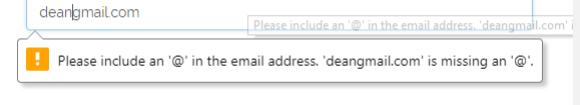
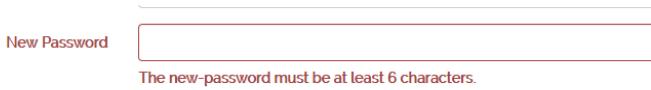
- Test Cases for Authentication Module

No	Test Case	Expected Output	Actual Output	Status
1	Provide wrong password	Getting error message and prevent from login.	  User Name 2014C015 Students should use student registration number as a user name. Ex: 2014C015 These credentials do not match our records.	Pass
2	Provide wrong username	Getting error message and prevent from login.	  User Name 2014C015 Students should use student registration number as a user name. Ex: 2014C015 These credentials do not match our records.	Pass
3	Too many login attempts	Temporarily disable login system for one minute after three login attempts.	  User Name dean Students should use student registration number as a user name. Ex: 2014C015 Too many login attempts. Please try again in 47 seconds.	Pass
4	Default password receiving from email	Default password should receive from the system to Lecturer,	  To: <madumaljeewantha@gmail.com> 11 days ago New result released To: <dasu@gmail.com> 11 days ago <b>Default Password - Results Management System</b> <b>To: &lt;parahaladan@gmail.com&gt;</b> 11 days ago Default Password - Results Management System To: <exam@mail.com> 11 days ago	Pass

		<p><b>AR and Examination branch users.</b></p> <p>Dear Prahalathan,</p> <p>This is your username and default password for Results Management System of Faculty of Management Studies &amp; Commerce, University of Jaffna.</p> <p>We highly recommend to change the default password after your first login. Have a great time!</p> <div style="background-color: #f0f0f0; padding: 5px; margin-top: 10px;"> <p>User Name : EMP001 Password : Ja9MNXg1KoZ2P5kJ66pY</p> </div> <div style="text-align: center; margin-top: 10px;"> <a href="#" style="border: 1px solid #0056b3; padding: 2px 10px; color: inherit; text-decoration: none;">Log In Now</a> </div>	
--	--	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

Table E.6 Test Cases for Authentication Module

- **Test Cases for Profiles Module**

No	Test Case	Expected Output	Actual Output	Status
1	Provide wrong email address	Should receive warning message		Pass
2	Provide correct details	Getting success message.		Pass
3	Provide wrong current password to change password	Get error message saying current password is wrong.		Pass
4	Provide less than six character as new password in change password	Get error message showing new password should be greater than six characters.	 	Pass
5	Provide not matching password to confirm password in change password	Provide error message and say password mismatch.		Pass

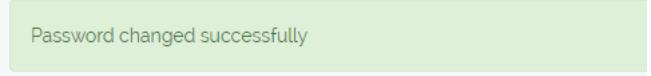
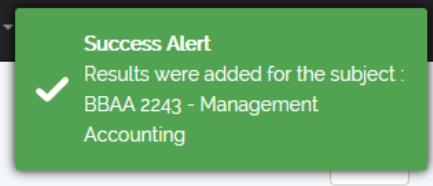
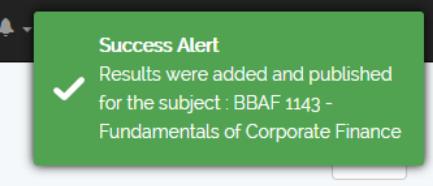
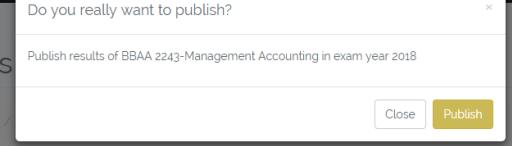
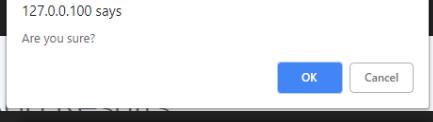
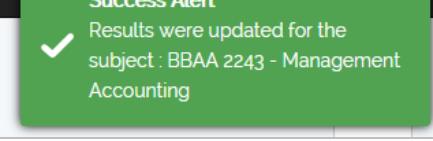
<b>6</b>	Provide correct details in change password.	Get success message and change password.		Pass
----------	---------------------------------------------	------------------------------------------	------------------------------------------------------------------------------------	------

Table E.7 Test Cases for Profiles Module

- **Test Cases for Grades Module**

No	Test Case	Expected Output	Actual Output	Status
1	Add new grades	Get confirmation and success notification.	  	Pass
2	Add new grades and publish	Get confirmation and success notification.	  	Pass
3	Publish results	Get confirmation and success notification.	  	Pass
4	Update results	Get confirmation and success notification.	  	Pass

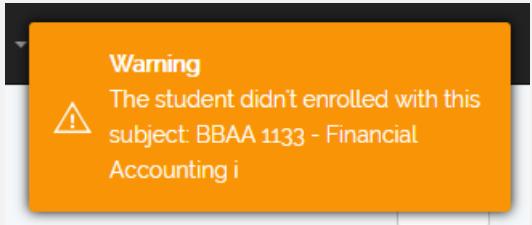
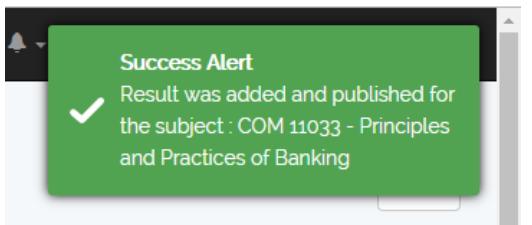
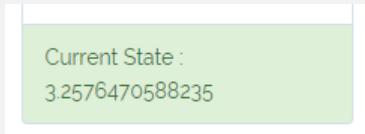
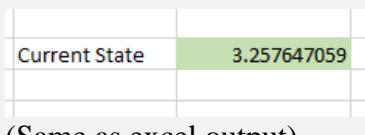
5	Add new repeated subject results to particular student who do not enrolled with particular subject.	Get warning message saying student not enrolled with particular subject.		Pass
6	Add new repeated subject results to particular student who enrolled with particular subject	Get success notification		Pass
7	Repeat subject handling; year 2013:C, 2014:MC, 2015:A	Algorithm should select final grade as B (Faculty may allow to repeat up to B)	Tested in JSON format	Pass
8	Repeat subject handling; year 2013:MC, 2014:A	Algorithm should select final grade as A	Tested in JSON format	Pass
9	Calculation of overall GPA	Should provide same as manual calculated GPA	  (Same as excel output)	Pass

Table E.8 Test Cases for Grades Module

- Test Cases for Lecturer Module

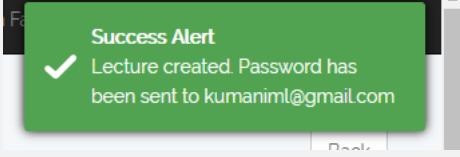
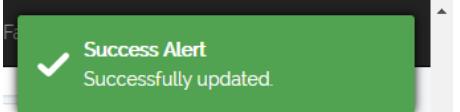
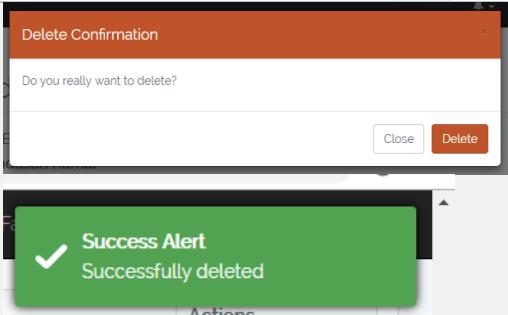
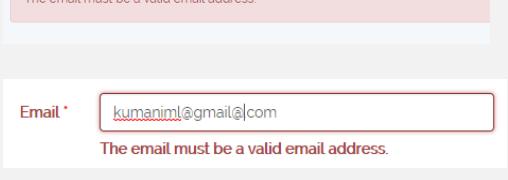
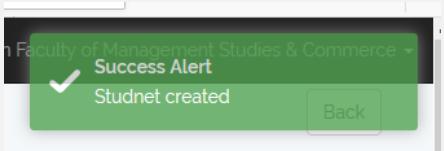
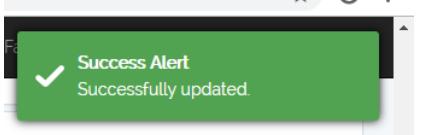
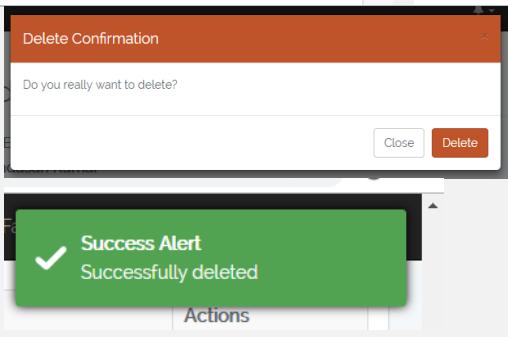
No	Test Case	Expected Output	Actual Output	Status
1	Add new lecturer	Get success notification.		Pass
2	Edit existing lecturer details	Get success notification.		Pass
3	Delete existing lecturer	Get confirmation window and success notification.		Pass
4	Enter existing employee number when adding new lecturer	Get error message and prevent from saving.		Pass
5	Enter wrong email address when adding new lecturer	Get error message and prevent from saving.		Pass

Table E.9 Test Cases for Lecturer Module

- Test Cases for Student Module

No	Test Case	Expected Output	Actual Output	Status
1	Add new student	Get success notification.		Pass
2	Edit existing student	Get success notification.		Pass
3	Delete existing student	Get confirmation window and success notification.		Pass
4	Add wrong email address while creating new student	Get error message and prevent from saving		Pass
5	Add existing student registration number while creating new student	Get error message and prevent from saving		Pass
6	Add wrong NIC number while creating new student	Get error message and prevent from saving		Pass
7	Add existing student	Get error message and		Pass

	index number while creating new student	prevent from saving	<p>Index No: * c14015 Exc14015 The student index number has already been taken.</p>	
--	-----------------------------------------	---------------------	---------------------------------------------------------------------------------------------	--

Table E.10 Test Cases for Student Module

- Test Cases for Subjects Module

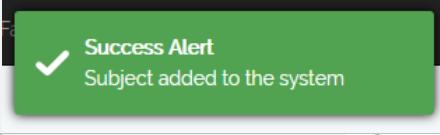
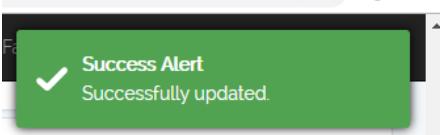
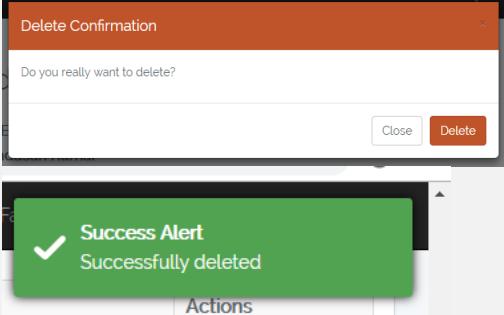
No	Test Case	Expected Output	Actual Output	Status
1	Add new subject	Get success notification.		Pass
2	Edit existing subject details	Get success notification.		Pass
3	Delete existing subject	Get confirmation window and success notification.		Pass

Table E.11 Test Cases for Subjects Module

- Test Cases for Reports Module

No	Test Case	Expected Output	Actual Output	Status
1	Students in specific academic year	Generate report with students in specific academic year	Report generated successfully	Pass
2	Students in specific course	Generate report with students in specific course	Report generated successfully	Pass
3	Students in specific department	Generate report with students in specific department	Report generated successfully	Pass
4	Students in specific specialized area	Generate report with students in specific specialized area	Report generated successfully	Pass

<b>5</b>	Full details of a student	Generate report with full details of a given student	Report generated successfully	Pass
<b>6</b>	Assigned subjects of a student	Generate report with assigned subjects of a student	Report generated successfully	Pass
<b>7</b>	All the lecturers in the faculty	Generate report with all the lecturers in the faculty	Report generated successfully	Pass
<b>8</b>	Lecturers in specific department	Generate report with lecturers in specific department	Report generated successfully	Pass
<b>9</b>	Assigned subjects of specific lecturer	Generate report with assigned subjects of specific lecturer	Report generated successfully	Pass
<b>10</b>	Full results of a student	Generate report with full results of a student	Report generated successfully	Pass
<b>11</b>	Subject results	Generate report with given subject results	Report generated successfully	Pass
<b>12</b>	Semester results	Generate report with given semester results	Report generated successfully	Pass
<b>13</b>	Course details	Generate report with course details	Report generated successfully	Pass
<b>14</b>	Grading details	Generate report with Grading details	Report generated successfully	Pass

*Table E.12 Test Cases for Reports Module*

# Appendix F – Code Listing

## Change Password of User

Following code segment was used to change password of the users.

```
//Method to handle password chaning
public function changePassword(Request $request) {

    if (!($hash::check($request->get('current-password'),
Auth::user()->password))) {
        // The passwords matches
        return redirect()->back()->with("error", "Your current
password does not matches with the password you provided. Please try
again.");
    }

    //strcmp - used string comparison method
    if(strcmp($request->get('current-password'), $request-
>get('new-password')) == 0){
        //Current password and new password are same
        return redirect()->back()->with("error", "New Password
cannot be same as your current password. Please choose a different
password.");
    }

    $this->validate($request, [
        'current-password' => 'required',
        'new-password' => 'required|string|min:6|confirmed',
    ]);

    //Change Password
    //Get the user instance
    $user = Auth::user();
    //encrypt new password using bcrypt
    $user->password = bcrypt($request->get('new-password'));
    $user->save();

    return redirect()->back()->with("success", "Password changed
successfully");
}
```

## Default Password sending Markdown

Following conde segment shows the customized markdown for the sending professional looking email through Laravel [16] markdown.

```

@component('mail::message')

# Congratulations!
<p><strong>Dear {{ $request['name'] }},</strong></p>

<p>This is your username and default password for Results Management System of Faculty of Management Studies & Commerce, University of Jaffna.</p>
<p>We highly recommend to change the default password after your first login.  
Have a great time!  
</p>

@component('mail::panel')
    <strong>User Name : {{ $request['user_name'] }}</strong><br>
    <strong>Password : {{ $request['password'] }}</strong>
@endcomponent

@component('mail::button', ['url' => 'http://127.0.0.100/login'])
Log In Now
@endcomponent

Thank you,<br>
{{-- {{ config('app.name') }} --}}
Dean<br>
Faculty of Management Studies & Commerce<br>
University of Jaffna
@endcomponent

```

## Handle User Login

Handling user login functionality is responsible for the following code segment.

```
/**  
 * Handle a login request to the application.  
 *  
 * @param \Illuminate\Http\Request $request  
 * @return \Illuminate\Http\RedirectResponse|\Illuminate\Http\Response|\Illuminate\Http\JsonResponse  
 */  
public function login(Request $request)  
{  
    $this->validateLogin($request);  
  
    // If the class is using the ThrottlesLogins trait, we can  
    // automatically throttle  
    // the login attempts for this application. We'll key this by  
    // the username and  
    // the IP address of the client making these requests into  
    // this application.  
    if ($this->hasTooManyLoginAttempts($request)) {  
        $this->fireLockoutEvent($request);  
  
        return $this->sendLockoutResponse($request);  
    }  
  
    if ($this->attemptLogin($request)) {  
        return $this->sendLoginResponse($request);  
    }  
  
    // If the login attempt was unsuccessful we will increment  
    // the number of attempts  
    // to login and redirect the user back to the login form. Of  
    // course, when this  
    // user surpasses their maximum number of attempts they will  
    // get locked out.  
    $this->incrementLoginAttempts($request);  
  
    return $this->sendFailedLoginResponse($request);  
}
```

## Notify Students Results Job Class

This is responsible for sending emails to the students regarding new results. This is using Laravel [16] queue to speed up the application.

```
<?php

namespace App\Jobs;

use Illuminate\Bus\Queueable;
use Illuminate\Queue\SerializesModels;
use Illuminate\Queue\InteractsWithQueue;
use Illuminate\Contracts\Queue\ShouldQueue;
use Illuminate\Foundation\Bus\Dispatchable;
//For Sending mail
use Illuminate\Support\Facades\Mail;
//Mailable file
use App\Mail\notifyStudentResultsMailable;

class notifyStudentResultsJob implements ShouldQueue
{
    use Dispatchable, InteractsWithQueue, Queueable,
SerializesModels;

    /**
     * Create a new job instance.
     *
     * @return void
     */
    protected $request;

    public function __construct($request)
    {
        $this->request = $request;
    }

    /**
     * Execute the job.
     *
     * @return void
     */
    public function handle()
    {
        Mail::to($this->request->email)->queue(new
notifyStudentResultsMailable($this->request));
    }
}
```

## Deans Only Middleware

This middleware is used to manage the Dean's access privileges. If the request is coming from user Dean, it will allow to proceed otherwise it will redirect to back.

```
<?php

namespace App\Http\Middleware;

use Closure;
use App\User;
use Illuminate\Support\Facades\Auth;
use Illuminate\Http\Request;

// Registered as "deansonly"
class DeansOnly
{
    /**
     * Handle an incoming request.
     *
     * @param \Illuminate\Http\Request $request
     * @param Closure $next
     * @return mixed
     */
    public function handle($request, Closure $next)
    {
        if (!Auth::check()) {
            return redirect('/login')->with('warning', 'Unauthorized page. Please login with your credentials');
        } else {
            //Get user type
            $type = Auth()->user()->type;

            if ($type != 'dean') {
                //redirect
                return back()->with('warning', 'Unauthorized page.');
            }
        }
        //Allow access to the request
        return $next($request);
    }
}
```

# Appendix G – Client Certificate

මගේ අංකය  
සමතු තිබ.  
My Number }

තමේ අංකය  
සමතු තිබ.  
Your Number }

දුරකථනය : 021-2222294  
තොත්‍යාපනය : 021-2222483  
Telephone : 021-2222006



නෑ.පෙ.ඩැය 57  
තිරුනේල්වෙලි  
යාපනය.

ත.ප.எ.ණ 57  
තිරුනේල්වෙලි,  
යාපනය

P.O.Box 57,  
Thirunelveli,  
JAFFNA

යාපනය විශ්වවිද්‍යාලය, ශ්‍රී ලංකාව  
යාපනයාපනය පල්කලාක්කழකම්, මිලංකා  
**UNIVERSITY OF JAFFNA, SRI LANKA**

1<sup>st</sup> October 2018

Project Examination Board,  
University of Colombo School of Computing,  
35, Reid Avenue,  
Colombo 7

Dear Sir/ Madam,

## Letter of Certification

I would like to inform you that Mr. S.M. Madumal Jeewantha (R160017) has successfully developed a Results Management System for the Faculty of Management Studies & Commerce, University of Jaffna. The project was undertaken by him as the final year project of Bachelor of Information Technology Degree in UCSC, University of Colombo.

I must appreciate the effort he has been taken to develop the system with providing important requirements of the faculty.

This certification is issued upon the request of Mr. S.M. Madumal Jeewantha.

Thank you,

Yours sincerely

  
Prof.R. Vigneswaran  
Vice-Chancellor

VICE CHANCELLOR  
University of Jaffna  
Jaffna, Sri Lanka

# Glossary

**AJAX** – Stands for Asynchronous JavaScript and XML. Combine collection of technologies. Running on Client side and helps to develop interactive web applications

**Apache** – Secure web server developed by Apache Software foundation.

**CSS** – Stands for Cascading Style Sheets. Use to apply styles for Markup languages such as HTML, XML.

**Laravel** – Laravel is a free, open-source PHP web framework, created by Taylor Otwell and intended for the development of web applications following the model view-controller architectural pattern and based on Symfony.

**HTML** – Stands for Hyper Text Markup Language. Use to build Web pages.

**Interface** – Interconnect web system with the user.

**JavaScript** – Use to dynamic web applications. Develop by Netscape.

**MySQL** – One of most popular Database management system can handle big amount of data related to different types.

**PHP** – Stand for PHP Hypertext Preprocessor. Object oriented supported server side scripting language.

**RUP** – Stands for Rational Unified Process. Iterative software development methodology. Develop by Rational Software Co-operation.

**WAMP** – Open source Bundled software package. Include Apache, PHP, MySQL.

**JQuery** – It is a JavaScript library. It was designed to simplify the client-side scripting of HTML.

**JSON** – Stands for Java Script Object Notation. Light weight data exchange format.

Have similar characteristics of **XML**

**SQL** - Stands for Structured Query Language. Help to retrieve data base details.

**UML** - Stands for Unified Modeling Language. Developed by Ivar Jacobson, James Rumbaugh, and Grady Booch at Rational Software. It is a modeling language.

# Index

## A

- Abstract ..... iii  
academic year ..... 39, 44, 52, 69, 77, 79, 84, 85, 87, 91, 99  
Analysis ..... 4, 16, 54  
Apache ..... iii, 26, 29, 55, 107  
AR 7, 43, 61, 62, 64, 65, 66, 70, 72, 73, 90, 93  
Assistant Register ..... xi  
Authentication ..... 40, 43, 70, 93, 94

## B

- Bootstrap ..... iii, 20, 21, 28, 38, 55

## C

- Commerce ..... iii, 1, 2, 7, 50, 54, 84, 86  
course ... 3, 9, 11, 24, 40, 41, 43, 44, 67, 68, 75, 76, 82, 84, 85, 88, 90, 92, 93, 99, 100  
CSS ..... iii, 28, 38, 55, 107

## D

- Dashboard ..... 4, 25, 39, 40, 81  
database ..... iii, 5, 11, 19, 29, 34, 59  
Dean ..... 24, 25  
design ... iii, 5, 14, 16, 19, 20, 22, 26, 28, 54

## E

- email. 23, 30, 36, 40, 65, 66, 92, 93, 94, 97, 98, 101

## F

- Faculty ..... iii, 1, 7, 10, 50, 54, 67, 96  
FMSC ..... 2, 3, 6, 10

## G

- GPAiii, 3, 4, 7, 9, 11, 31, 34, 44, 50, 61, 96  
Grades ..... 40, 44, 95, 96

## J

- Jaffna ..... iii, 1, 50, 54  
JavaScript ..... iii, 28, 38, 39, 55, 107

## L

- Laravel.... iii, 27, 30, 34, 36, 38, 39, 40, 52, 55, 56, 59, 101, 104, 107  
lecturer iii, 3, 10, 43, 44, 45, 47, 48, 73, 74, 82, 87, 90, 97, 100  
Login ..... 10, 23, 43, 103

## M

- Management . iii, 1, 2, 7, 17, 18, 29, 50, 54, 57, 58, 60, 84  
Module....10, 11, 24, 39, 40, 41, 43, 44, 45, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100  
MySQL ..... iii, 26, 29, 55, 57, 107

## P

- PHP ..... iii, 26, 27, 38, 55, 57, 107

## S

- semester ..... iii, 9, 31, 41, 44, 76, 88, 100  
Student iii, 9, 11, 34, 54, 61, 63, 70, 72, 73, 80, 81, 82, 84, 98, 99  
Subjects ..... 41, 43, 45, 69, 76, 99  
System ..2, 3, 4, 7, 8, 10, 11, 14, 15, 17, 18, 20, 21, 22, 23, 27, 29, 42, 43, 50, 54, 57, 58, 59, 60, 63, 64, 65, 70, 82, 89, 92

## U

- UML ..... iii, 4, 15, 18, 54, 108  
University ..... i, iii, 1, 50, 54

