



STUDENT MANAGEMENT INFORMATION SYSTEM FOR NATIONAL INSTITUTE OF SPORTS SCIENCE

**A dissertation submitted for the Degree of Master of
Information Technology**

**P.H.A.U. De Silva
University of Colombo School of Computing
2018**



Declaration

The thesis is my original work and has not been submitted previously for a degree at this or any other university/institute.

To the best of my knowledge it does not contain any material published or written by another person, except as acknowledged in the text.

Student Name : P.H.A.U. De Silva

Registration Number : 2015/MIT/053

Index Number : 15550537

Signature:

Date:

This is to certify that this thesis is based on the work of ~~Mr~~ /Mrs P.H.A.U De Silva under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by

Supervisor Name : DR. K. L. Jayarathna

Signature:

Date:

Abstract

National Institute of Sports Science is the premier institute of Sports education in Sri Lanka. It was established under the Special Sports Act in 1973.

This report describes the analysis, design and implementation of a prototype of an information technology solution provided to the National Institute of Sports Science (NISS). The solution is mainly consisting with online web system and the documentation following its design, implementation, processes and its entities. Also this document consists of the Introduction to the system, Background of the study, Feasibility study and requirements Gathering, Requirement gathering techniques, Design, Introduction to the object oriented methods and Structured system analysis methodology. Implementation of the system, Testing, Maintenance, Critical Evolution and conclusion and the referencing that follows.

Acknowledgments

I would like to express my deepest gratitude to Dr. K .L. Jayarathna, Senior Lecturer, School of Computing, University of Colombo for his valuable guidance, kind hearted co-operation and encouragement extended throughout the study.

I wish to thank Dr.D.M.R.B.Dissanayake, former secretary to the Ministry of Sports and Mr. G.L.Sajith Jayalal, Director, National Institute of Sports Science for providing me the financial support from institutional funds to follow this postgraduate degree program.

Finally, I appreciate everybody who helps me in making this effort a success.

List of Acronyms

OOP	-	Object Oriented Programming
DBMS	-	Database Management Systems
DB	-	Database
UML	-	Unified Modelling language
RAM	-	Random Access Memory
IDE	-	Integrated Development Environment
NISS	-	National Institute of Sports Science
IS	-	Information System
UI	-	User Interface
EF	-	Entity Framework
MVC	-	Model View controller

Table of Contents

Declaration.....	i
Abstract.....	ii
Acknowledgments	iii
List of Acronyms.....	iv
List of Figures.....	viii
List of Tables.....	x
Chapter 1 : Introduction.....	1
1.1 The Problem statement.....	2
1.1.1. Project Objectives.....	3
1.2 Scope of Project.....	4
1.3 An outline of remaining chapters	6
Background Chapter	7
Design Chapter	7
Implementation Chapter	7
Testing Chapter	7
Conclusions	7
Chapter 2 : Background.....	8
2.1 Similar Systems	9
Chapter 3 : Analysis and Design/ Methodology.....	12
3.1 System Analysis	12
3.1.1 Current Manual System.....	12
3.2 Feasibility Study and Requirement Gathering.....	15
3.3. Requirement Specification	16
3.3.1 User interface.....	16
3.3.2 Hardware Requirements	16
3.4 User Groups.....	17
3.5 Functional and Non-Functional Requirements.....	18

3.5.1 Functional Requirements User Group – System Administrator	18
3.5.2 Functional Requirements User Group – Registered User	22
3.6.2 Non-Functional Requirements.....	25
Chapter 4 : Design/Methodology	27
4.1 Methodology.....	27
4.1.1 Agile Methodology.....	27
4.1.2 Features of Agile Model	28
4.2 Design.....	28
4.2.1 Architecture	29
4.3 Design Pattern Repository Pattern.....	30
4.4 Use Case Identification.....	32
4.5 Sequence Diagram.....	33
4.6 Database Design	35
Chapter 5 : Implementation	36
5.1 Introduction	36
5.2 Good practices followed during the implementation	36
5.3 Implementation environment and Development tools.....	37
5.4 Code and module structure description	38
5.4.1 Barcode Generation	41
Chapter 6 : Testing and Evaluation	42
6.1 Importance of testing.....	42
6.2 Selected Testing Methods.....	42
6.2.1 White Box Testing.....	43
6.2.2 Black Box Testing	43
6.2.3 Acceptance Testing.....	43
6.3 Test Plan and Test Cases	43
6.3.1 Test Plan	43
6.3.2 Test Cases	48

6.3.3 Acceptance Testing.....	54
Questionnaire Form for User Feedback.....	54
Chapter 7 : Conclusion	60
7.1 Introduction	60
7.2 Lessons Learned During the Project.....	60
7.3 Critical Assessment of the project	62
7.4 Limitations and Future Recommendation	63
Chapter 8 : References.....	65
Appendix A – User Documents.....	66
Appendix B – Reports	75
Appendix C – Test Cases	79

List of Figures

Figure 1.1 Scope of Project	4
Figure 2.1 Dashboard of student management system is OpenSIS	9
Figure 2.2 Dashboard of student management system is OpenSIS 11	10
Figure 2.3 Dashboard of Gibbon student management system	10
Figure 4.1 Use Case Diagram.....	32
Figure 4.2 Sequence Diagram of admin -user function.....	33
Figure 4.3 Sequence Diagram of administration login.....	33
Figure 4.4 Sequence Diagram of admin- course details.....	34
Figure 4.5 Implemented Database Design.....	35
Figure 5.1 use of Encapsulation of project	39
Figure 5.2 Use of inheritance	40
Figure 5.3 Important codes of NISS barcode generation function	41
Figure 5.4 Sample Barcode	41
Figure 6.1 Employee Suggestion of System Design	59
Figure 8.1 Login Form	66
Figure 8.2 Login Form with Validation Messages	66
Figure 8.3 Login Form Enter Details	67
Figure 8.4 Invalid Credential Message.....	67
Figure 8.5 Screen Shot of Create Form Resource Type	68
Figure 8.6 Screen shot of Resource Type Model Class.....	68
Figure 8.7 Resource Type Index Page.....	69
Figure 8.8 Screen shot of Resource Type Index Controller	69
Figure 8.9 Screen Shot of Resource Type create controller Method.....	69
Figure 8.10 Screenshot of successfully resource type record added	70
Figure 8.11 Screenshot of editing records	70
Figure 8.12 Screenshot of delete view resource type	71
Figure 8.13 Screenshot of Detail View	71
Figure 8.14 Form Staff Users	72
Figure 8.15 Screen shot of Create Staff Controller Method.....	72
Figure 8.16 Screen Shot of Staff Index	73
Figure 8.17 Screen Shot of Staff Index Controller.....	73
Figure 8.18 Staff Create form with Inputs.....	73
Figure 8.19 - Screenshot of Staff Edit Controller Method	74
Figure 8.20 - Screenshot of Delete View Staff.....	74

Figure 8.21 Screen shot of Personal Details Report, No of Students, Monthly Attendance	75
Figure 8.22 Screen shot of Course Schedule Report, Resource Allocation	76
Figure 8.23 Screen shot of Gender wise user report.....	77
Figure 8.24 Screen shot of User Type Report	78

List of Tables

Table 2.1 Existing System Comparison	11
Table 3.1 Function Login	18
Table 3.2 Function View, Add, Delete	19
Table 3.3 View and handle pending student registration	20
Table 3.4 Manage student records	20
Table 3.5 Sending e-mails to students and staffs.	21
Table 3.6 Search, Delete, block, unblock, update or Assign user to different roles.....	21
Table 3.7 Send Newsletters to registered users	22
Table 3.8 Handle Payment Gateway	22
Table 3.9 User Group – Register User	23
Table 3.10 Login with change and forgotten password.....	23
Table 3.11 View Schedule.....	24
Table 3.12 View Courses	24
Table 3.13 View Exam Details.....	25
Table 3.14 View Resource Details	25
Table 4.1 MVC architecture	30
Table 6.1 Test Plan for Student Management System for NISS 1	44
Table 6.2 Test Plan for Student Management System for NISS 2	44
Table 6.3 Test Plan for Student Management System for NISS 3	45
Table 6.4 Test Plan for Student Management System for NISS 4	46
Table 6.5 Test Plan for Student Management System for NISS 5	47
Table 6.6 Test Case Login.....	49
Table 6.7 Summary of testing 1	51
Table 6.8 Summary of Testing 2	53
Table 6.9 Summary of Testing 3	80
Table 6.10 Weighted Value 1	56
Table 6.11 Responses for the Likert Scale	57
Table 6.12 Mean Value Calculation	58
Table 6.13 Analysis of system design	58
Table 7.1 Successfully rate of the project.....	62
Table 7.2 Critical Assessment of the Project.....	63

Chapter 1 : Introduction

Organization performance to achieve its goals and objectives depends on the resources available both inside and outside also numerous resources are required to perform that include manpower, finance, materials, facilities and information. Information is one of the most vital resource of any organization that have an impact on all the other resources. But those resources are not enough without having timely acquisition, utilization and monitoring which are utterly essential for the success of the organization. Therefore, it is mandatory to have an accurate system in any organisation to handle recourses accurately.

National Institute of Sports Science (NISS) is the premier institute of sports education in Sri Lanka which conducts more than ten different types of sports educational courses per annum. Course duration is one year or less than one year and depending upon category of each course. However, every year 1000 students are passed out as sports graduates. Even though this institute is having such a courses every year, all documentations are being defended completely on a manually system, therefore accuracy of information as well as decision makings are get delayed due to this manually system that they are working with. As a result of mentioned key factors, success of the organization is being directly affected and will be unable to have any modification to the system that they depend on.

The vision of NISS is “To be centre of excellence in Sports education in South Asia”.

However, to obtain the maximum outcome from NISS, a system implementation need to be introduced to gain following benefits to direct organization to the accomplishment.

- Operations

Information systems can offer more complete and more recent information, allowing to operate institute more efficiently. Also can use information systems to gain a cost advantage over competitors or to differentiate NISS by offering better customer service.

- Decisions

The NISS information system can help make better decisions by delivering all the information need and by modelling the results of decisions. A decision involves choosing a course of action from several alternatives and carrying out the corresponding tasks. When NISS has accurate, up-to-date information, it can make the choice with confidence.

- Records

NISS needs records of its activities for student evaluation, examination evaluation, and certificated issuing as well as for finding the causes of problems and taking corrective action. The information system stores documents and revision histories, communication records and operational data.

- Real Time Data

Through having integrated information in one system, all of the data is up to date. This is essential for all elements of the organization successes and offer better service.

- Better Communication

Team members will be able to communicate better through having exactly the same information available to them at each time. Its saves having to miss-match data between systems and departments.

- Reduced Risk of Errors

Due to the fact that data will not have to be replicated, there is less chance of human errors being made which leads to more accurate information available.

- Greater Productivity

Employees can spend more time on tasks that will help the organization to grow, rather than having to replicate data and wait for information to be sent to them from other departments.

- One Secure Location

Through having all of the necessary data stored in one information system, relevant data is easier for employees to access.

These software systems enable to supervise student-related activities such as keeping records of tests or examinations conducted, attendance, appraisal on performance including details of marks scored, particulars of everyday attendance, and all other institution-related activities. This Management information system has diverse application potentials ranging from management of all student-related functions as well as administrative functions of a NISS.

1.1 The Problem statement

According to the current proceedings NISS has a difficulty of keeping records, operational information and validation of certificates. Therefore, erroneous result can be seen due to their

manually day today functions and NISS requires to avoid these issues that are being currently faced in efficient manner. Management Information System (MIS) is a vital factor to be introduced at earliest to advantage improvement of employee's satisfaction, managing student's records, generating of accurate reports. On the other hand, by using a mobile app any authorized student or user can be accessed their information system and make own decisions according to the data. Also another concern NISS facing is managing the courses schedules as well as other resources involving for conducting all academic courses. NISS has only limited resources to manage their current courses that are being conducted, therefore there should be a proper management of scheduling & allocation of resources and shortage of such a system is currently facing a one of the biggest problem in NISS. Also updating of course details, modifications and conveying urgent news on courses to student are currently having a problem because, it is finding difficult to get student data (such as contact number) immediately and keep they informed instantly. Also NISS is currently receiving certificates from various third party organizations to certify & confirm validation as there are fake certificates in society and NISS are facing biggest issue to find oldest data due to their manual system. Therefore incorrect signed certificates can be released to student.

1.1.1. Project Objectives

- Provide better and improved customer care solution

One of the main objectives of the system is to provide the best possible customer service and uplift the customer satisfaction.

- To increase the efficiency of the National Institute of Sports Science

With the new system being implemented it is expected to increase the efficiency of the National Institute of Sports Science by automating the functions of the National Institute of Sports Science.

- To provide a user friendly environment by adopting technology.

With innovation of technology, it is impossible to neglect the impact of technology to day today business. So by adopting the new system it will provide the edge of technology and create an environment that the users will enjoy.

- Minimize the manual labour

The automated system will allow the restaurant to minimize the manual workload by transforming the workload to the system.

- Reduce time waste by maximizing resources

The system is expected to function with maximum efficiency, thereby it is expected to speed up the process.

1.2 Scope of Project

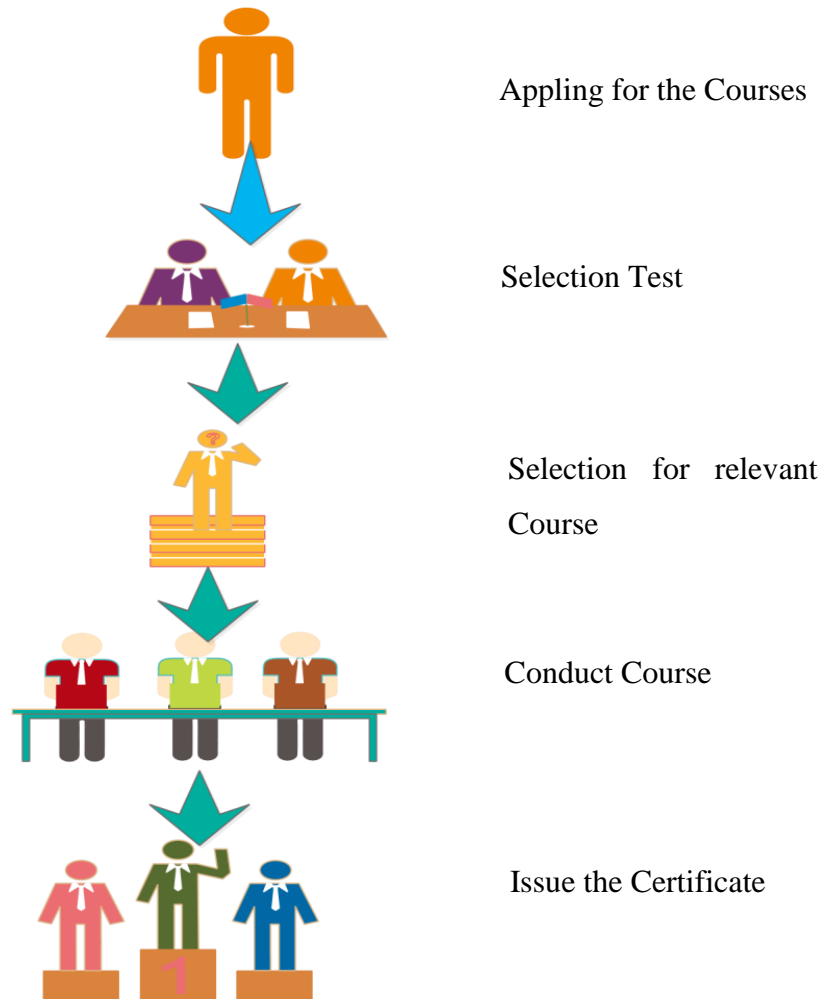


Figure 1.1 Scope of Project

1.2.1 Features of Systems

1. Admission of students to the institution

By using online application form student are allowed to apply for course as they desire. Also during the time of applying in certain courses, relevant course coordinator will get an instant notification. Then, the course coordinator is supposed to check qualifications of the student and proceed with sending an admission letters for the selection test that would be written/practical exam or sending regret letters for non-qualified students. After the selection exams is being

passed, enrolling process of those student to relevant to be implemented. As soon as enrolment is done, students are allowed to get information on their course details, offered hostel &, library facilities as well as details of relevant academic staff.

2. Availability of information from a single source

Use of such online administrative and student information systems increases the functional efficiency of an institution and improves timely decision-making processes at all levels. Availability of a student records system containing all information details at a single source enables easy percolation of the right information.

3. Centralised information sourcing and management

Although the very objective of this intelligent student information management system is to tackle various issues related to students, administration and lectures, it offers a wider perspective by providing a complete information management solution. The centralised e-Advising offered by the current intelligent student information systems allows the students to send a preregistration email to register. The web link included in the reply mail enables them to access a complete academic planning network which includes a gamut of information about various programmes, courses, fee structure, further progress, and other employment openings. Thus, students can compare the programmes and facilities available in various other institutions and make a choice on preferences. Current student information systems are designed in such a way as to provide all the comprehensive information in not more than three screen levels with customised applications and a single log in. This student information management system efficiently integrates all the information management functions of an institution right from the students' level up to the level of the Registrar's office. It even enables institutional study and forecast.

4. Centralised accounting and billing procedures

This student information system enables all accounting and billing functions while simultaneously managing the academic requirements of the students and the administration.

5. Monitoring student-related activities

A complete record of students' attendance and leave details is stored in the system. The reminder option in the system informs the institution management about the irregularities in the attendance or leave details for further action. This system offers a complete follow-up on all the discipline records of the students.

6. Easy-to-use updated process

This information system enables compiling of all the data at as a single source, especially for students and lectures. This electronic storage of records is a good solution for replacing the manual record storing method. Such an availability of all the required information in the electronic form facilitates easy retrieval, and also speeds up approval formalities. It is possible to store all student-related activities such as attendance, test performance, and exam schedules in the system and share these with the parents and other faculties.

7. Easy scheduling of examinations

Scheduling of examination dates can be easily handled by a student information system. It correlates all details such as availability of lectures and completion of book syllabus fixed for the term before announcing the examination dates.

Easy-to-decipher reporting functions for individuals as well as departments, and facilitate the generation of real-time reports and customised reports.

Flexible to operate in multiple ways with easy-to-change operating or processing setups, in tune with current requirements.

Easy integration with the other modules that already exist; also offer ingenuity during integration.

8. Reports & Analytics

Following are the reports will include to the system

1. Name list of applied individuals for the specific courses
2. Name list of selected students for each and every courses
3. Attendance report
4. Exam result
5. Detailed certificate
6. Final certificate of course with barcode
7. Lectures list
8. Courses that were being conducted annually.

1.3 An outline of remaining chapters

This section provides brief description on next chapters of the thesis. Each sections describes different activities which were carried out during the project.

Background Chapter

The background chapter provides the general idea of the existing similar systems to NISS Information System and the comparison among the NISS Information System and existing similar systems. And drawbacks of the existing system although those drawbacks how adapt in the NISS Information System. Also background chapter provide detailed literature review of features and components should consist.

Design Chapter

The design chapter provides detailed description about designing of various Unified Modified Language (UML) diagram and User Interface (UI) designed for this NISS Information System project.

Implementation Chapter

The implementation chapter provides available implementation techniques, methods and tools. Also detailed description of selected methods and techniques with justifies reasons for select those methods and techniques.

Testing Chapter

The testing chapter is providing details testing conducted to the NISS Information System and evaluation of the NISS Information System. Also provide different test cases and testing techniques used to test the NISS Information System.

Conclusions

The conclusion chapter is for this dissertation critically review the work carried out during the project duration.

Chapter 2 : Background

National Institute of Sports Science (NISS) is the premier institute of sports education in Sri Lanka. NISS control by local government and it is non-profitable institute. It has several responsibilities such as

- To provide sport education for sports sector personnel.
- To update knowledge of sports sector personnel.
- To register all coaches in Sri Lanka.
- Act as a sports educational facilitator of Sri Lanka.
- To provide facilities for the general public to determine the percentage of body fat, water & muscle, BMI (Body Mass Index), ideal weight level as per height and waist to hip ratio etc.
- Nutritional Advices and exercises recommendations are also provided free of charge.
- To provides Sports Library facilities

Annually more than 1000 students are qualified as sport graduates after following below mentioned courses.

1. Advanced diploma in Sports science
2. Diploma in Sports Science
3. Physical fitness instructor training courses
4. Coaches registration program
5. Certificate courses in sports science
6. Certificate courses in Yoga
7. Injury management course
8. Information technology for sports sector

Currently NISS has more than 5000 students who passed out as sport graduates in sports sector of Sri La

nka and at present, their data are being stored in manual filling system. Any organization all over in Sri Lanka who wants to carry out any sports related recruitment, need to get a verification of certificate from NISS. Therefore, due to their on-going manually system, organisation is facing huge waste of time on these day today affaires.

2.1 Similar Systems

As the initial step went through a rigorous search on the web to find out other student management systems. Found good result regarding the search. There are very powerful student management systems developed by various types of companies in the world. There are systems with powerful functions and extraordinary systems on internet. Institute owners or responsible persons can purchase the software directly from the internet.

First student management system is OpenSIS and this system currently available in lot of schools in United States of America. This system got more than required scope availability and this system is designed for USA education system. This system is available 15-day free trial period to test and decide whether buy or not. OpenSIS solution dashboard figure provided under Figure 2.1.

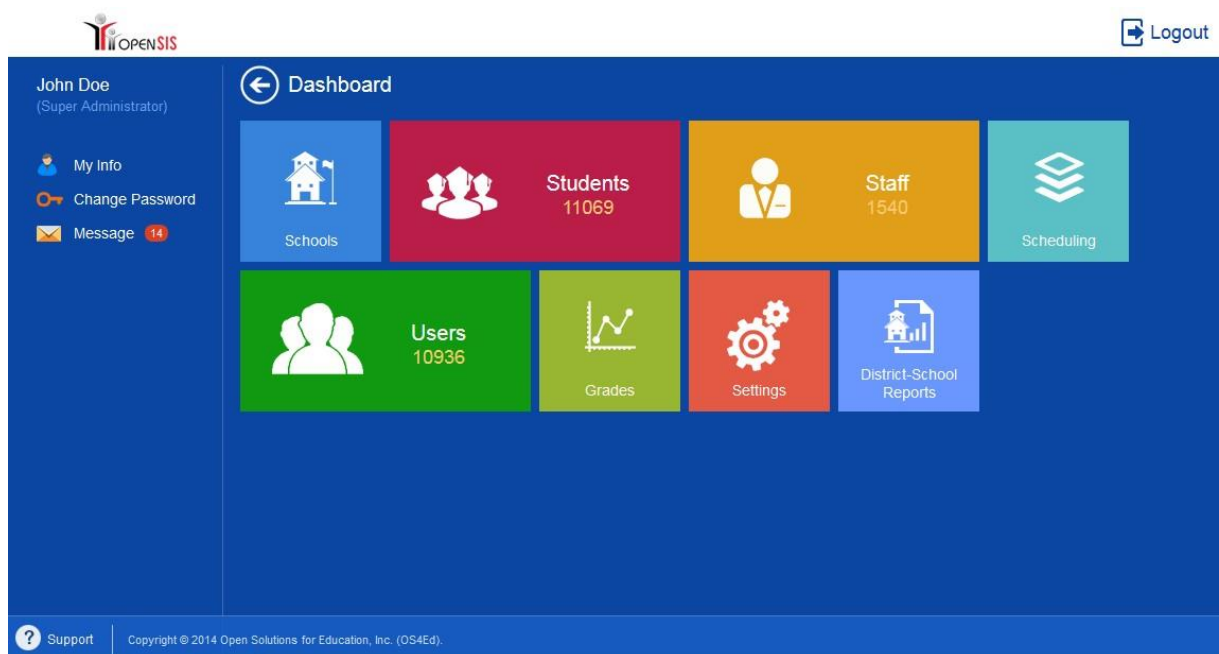


Figure 2.1 Dashboard of student management system is OpenSIS

Next system is Project Fedena is the open source school management system based on Ruby on Rails. It was initially developed by a team of developers at Foradian Technologies. The project was made open source by Foradian, and is now maintained by the open source community. This system also currently available on internet. This is initially open source project but currently this system developed as commercial project.

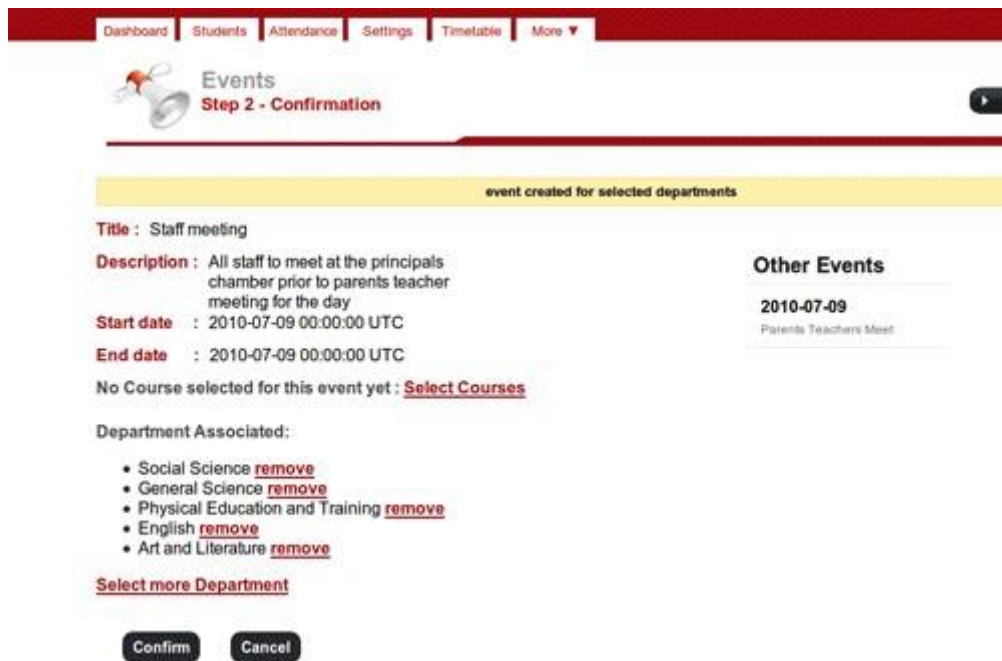


Figure 2.2 Dashboard of student management system is OpenSIS 11

Next system is Gibbon student management system. This system also currently available on internet to buy as commercial system. This system also has more than required scope availability. Lot of unnecessary features are available in this system. This system currently implemented in lot of institutes and schools all around the world.

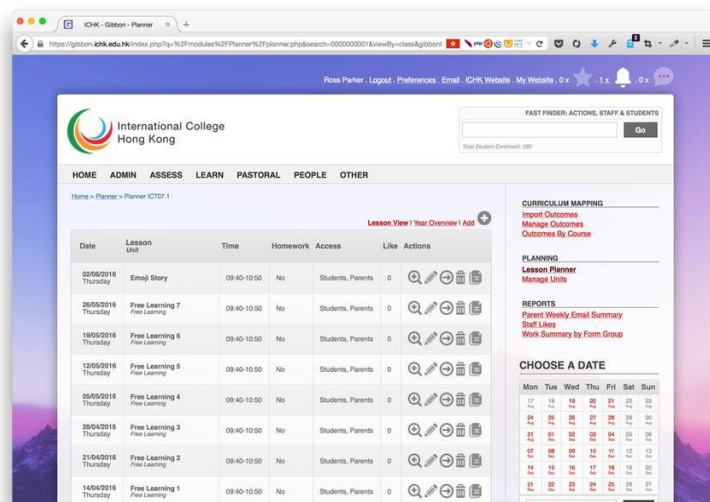


Figure 2.3 Dashboard of Gibbon student management system

Feature	Implemented System (Student Management System for NISS)	Fedena	Gibbon
Access from anywhere	Yes	Yes	No
Multiple User Interface	Yes	Yes	No
Platform Independence	Yes	No	Yes
High Expenditure	No	Yes	Yes
Requirement satisfaction	Yes	Yes	Yes
Categorisation	Yes	Yes	No

Table 2.1 Existing System Comparison

According to above Table 2.1 comparing to other existing systems with Student Management System for NISS current existing systems has some drawbacks but Fedena system and Gibbon DB is not free product user has to expend more money to buy them also Student Management System for NISS has all features of these existing system with better improvement.

Chapter 3 : Analysis and Design/ Methodology

3.1 System Analysis

3.1.1 Current Manual System

National Institute of Sports Science (NISS) currently use manually system for complete their day to day tasks. NISS use manually registration system to register new students. In this case when students want to register to NISS courses they have to fill the forms manually. NISS provides different type of courses therefore each course has different type of registration forms to fill. Then Received registration form examine by Course Coordinator. Course Coordinator is informing the selected students. Then students are face the written and practical exam. After that students who pass the exam will be enrol to the course.

Enrolled students attends is taking by manual daily log. Also student's appraisal and exam results announcing done by manual spreadsheet or manually paper work. Exam marks is generated by also using spreadsheets. Also student's certificate also is generated by using mail merge method also keep the certified student list manually.

NISS currently take decision by looking at manual records that they are keep in recent. When analysing manual report, it consumes the considerable time. Also they allocate their resources like class rooms, computer laboratory and library etc. And keep records of resources done by manually. Also NISS provides various courses those courses schedules also done by manually too. Currently all process of NISS handle by manually.

3.1.1.1 Drawbacks of current system.

National Institute of Sports Science has following drawbacks in their current manual system.

1. Consuming time analysing manual records.
2. Manual student registration form examines take time.
3. Manual Records data may inaccurate in time to time
4. Records can be misplaced.
5. Too much of books or document to handle.
6. No proper certificate validation

3.1.2 Proposed Solution

Proposed solution has main two parts.

1. Web Information System.
2. Supporting Mobile Application

3.1.2.1 Web Information System

Web System is going to handle the all processes of National Institute of Sports Science. Web System is manage all data and associated with admissions, course management, scheduling, registration, student record management, course resource allocation, current course progress, exam management and grading, certificates issue with barcode and report generations.

3.1.2.2 Manage Data and Associated Admissions

Web information system is proposed to handle all data of National Institute of Sports Science. All data of National Institute of Sports Science (NISS) is proposed to create, update, retrieve and delete using web information system. Also created data will be stored in Database Management System Server. Data management proposed to use Microsoft SQL Server.

Also web information is not allowed every user access to every data. This web system handles the admissions or access privileges. In that case specific users only allowed to all access of data and handle data. According to their associated admissions access to data and manage data permissions will be handled.

3.1.2.3 Course Management

National Institute of Sports Science (NISS) providing different courses registration of students proposed handle using web information system. Courses registration form will be displayed on NISS web site students have to complete and submit in online. Student's data will be saved in NISS web information system. Stored data will accessible to the course coordinator. Course Coordinator is select students using web system to access data and provide different requirement different courses. Web information system handle the student's notification through emails.

In course management web system is proposed to handle creating courses and handle or assign lecturers to courses. Also web system is proposed to handle the courses scheduling. In course creation course coordinator or course handling lecturer design their courses using web information system. Every course details will have stored in web information system database server.

3.1.2.4 Schedule Management

National Institute of Sports Science has schedule to manage. In that case web information has proposed function to manage the schedules of courses, resource allocation and lecturing schedules. Web information system will create, update and delete the schedule of National Institute of Sports Sciences.

3.1.2.5 Resource Allocation

National Institute of Sports Science has different resources. Handling details of resource details is important therefore proposed web system has resource allocation function to accomplish this task. In this case all resource details will be stored in database of web information system.

Resources allocation will have done according to the schedule of resource allocation.

3.1.2.6 Progress Monitoring

In progress monitoring is majorly to propose monitor the course's ongoing process. Web information system keep track of students and courses. That will provide the progress of current ongoing courses.

3.1.2.7 Exam Management and Grading

Exam is created by lectures. Managing exam is going to handle by proposed web information system therefore all exam details and schedules will store in database of web information system. Also grade of students publish using web information system.

3.1.2.8 Certificate Issue and Barcode Generation

National Institute of Sports Science decided to add more authorization to their certificates using barcodes. Barcode will have generated using by proposed web information system. Also certification issue also manage by web information system.

3.1.2.9 Report Generation

Data analysing is very important to the therefore all data record by web information system will help to generate report required by National institute of Sports Science.

3.1.2.10 Mobile Application

Mobile application is support to enable more accessible to user's system of National Institute of Sports Science.

3.2 Feasibility Study and Requirement Gathering

Requirement gathering is the process of identifying detailed description of functions that a newly proposed software solution should provide.

Properly gathered requirements plays a vital role for software solution to become a success and if not identified properly, the whole project could end up as an utter failure. Designing a system is fully based on the gathered requirements. There are number of techniques available to gather requirement for a proposed system. These techniques are called as fact finding methods. The most widely known fact finding methods are

- Questionnaires,
- interviews,
- Research
- Site visits
- Sampling of existing documents.

Among them Questionnaires, Interviews and Site-visits were used as the core requirement gathering techniques for the implemented web based information system solution for National Institute of Sports Science. As the first requirement gathering technique, a questionnaire was distributed among the existing members of the society along with some random animal lovers who are intended join with organization. Each person was asked to answer the questionnaire independently.

As the second requirement gathering technique, few interviews were carryout by selecting three working members from the National Institute of Sports Science. There are four types of interviewing methods such as structured, unstructured, close ended and open ended interviews. Among them, structured type interviews were carried out since they are the most informative interview type among all available types.

Web site visit method was used to gathering requirements and few number of relative web sites are visited for gathered information. All web sites belongs to various internationally recognized organizations and they all adopt to information system concept as their core working method.

After carrying out the all the above mentioned fact finding methods, requirements that should be fulfilled by the proposed online web information system solution for National Institute of Sports Science were verified and allocated into different sections as follows.

3.3. Requirement Specification

3.3.1 User interface

According to the gathered requirements user interface should consist a fixed theme through the entire system. This could be achieved by using Master page concept in ASP.NET technology. All users of the system (online solution) should have the capability to navigate from page to page by pressing buttons on the ribbon and the whole interface should follow a proper and clear structure. To reduce the number of errors due to incorrect data and excessive text inputs the users should be given predefined options like check boxes, radio buttons, and scroll ups and downs. If the user tries to insert unnecessary or irrelevant information or data, error messages should be displayed on the relevant place he or she incorrectly entered whereby they can understand how to resolve the problem in a correct manner. The whole system should be displayed by using English language (UK standard) and currency figures should be displayed in Sri Lankan rupees. The whole system should be self-explanatory for any user under the right age category.

3.3.2 Hardware Requirements

Server side

- A Desktop or a laptop computer with the capability host a web application and a Database server.

Client side

- Desktop or laptop computer with ADSL internet connection. (anything above ADSL would be fine)
- Any mobile device such as Smart phone and tabs with internet facility.

Software requirements Server side

- Windows 7 ultimate operating system or higher
- Visual studio 2015 as the integrated development environment with C# ASP.NET development enabled.
- Microsoft SQL server 2014
- Internet information system 7 or higher as the web server. (IIS)

Client side

- Any OS such as Windows, Apple IOS, Ubuntu, Android.

- Desktop and laptops should have a web browser such as Internet explorer, Firefox, Chrome.
- Any inbuilt web browser for mobile devices.

3.4 User Groups

This constructed online web information system solution consist four user groups namely as Students, Lecturer, Visitor and Administrator. Each user group is capable to perform the following tasks.

User group – Students

- Course search – Student is capable to search course based various, subjects, categories and etc.
- Dynamic forum – Registered Students are capable to create thread or contribute into existing thread in the dynamic forum section (with regards to various stray related topic).
- Payment gateway Student is capable to direct them self to a payment sections where he or she will able to make their necessary payment their courses.
- User account creation and Management – All potential registered students can manage their account which is compulsory action for them to contribute, participate and payment things from the site.
- Course review insertion and recommendation for users – Registered students are capable to insert course reviews and they will able to find out about recommended courses.

User group – Administrator

- Adding and removing courses
- Modifying existing courses information (name and description)
- Viewing the list of subject categories that belong to a courses
- Adding and removing subject categories
- Editing existing subject categories' information (name and description)
- Viewing the list of courses in a specific subject category.
- Editing subject details, such as the subject's name, description.
- Assigning an existing category to an additional category (a course can belong to multiple categories) or moving it to another category
- Removing a course from a category

- Deleting a category from a category list
- Allowing administrators to access the courses, category, or resources administration pages right from the catalogue.
- Site administrator can view and handle pending students.
- Student details management capability for Admin such as access student course data.
- Updating the status of a course
- Setting credit card authentication details
- Setting the course start date
- Sending e-mails to students and lecturers
- Retrieving course details and the students address
- Manage registered user by deleting, adding them into different roles or etc.
- Manage articles, forum and send news letters to all registered users.

3.5 Functional and Non-Functional Requirements

Functional Requirements are those which are related to the technical functionality of the system.

Non-Functional requirements are requirements that specifies criteria that can be used to judge the operation of a system in particular conditions, rather than specific behaviours.

3.5.1 Functional Requirements User Group – System Administrator

Function Login

Login is functional requirement for National Institute of Sports Science Student Management System. Table 3.1 depicts the details of login functional requirement.

Function :	Login
Summary :	The system should allow the System-Admin to successfully login once the validated username and password are submitted.
Input :	Username and password
Process :	The system will check with the MSSQL server for a valid login. If the provided username and password are valid, the main home page applicable to the user is displayed, else the user will be denied of access.
Output :	Successful login should redirect to home page else error message for invalid login.

Table 3.1 Function Login

Function: View, Add, Delete and Update Resources, Students, Staff, Courses and Exams.

View, Add, Delete and Update Resources, Students, Staff, Courses and Exams are functional requirements for National Institute of Sports Science Student Management System. Table 3.2 depicts the details of this functional requirement.

Function :	View, Add, Delete and Update Resources, Students, Staff, Courses and Exams.
Summary :	The system should allow the users to successfully login once the validated username and password are submitted.
Input :	Details of resources, students, staff, courses, and exams filled into required fields in web system.
Process :	The system will check with the MSSQL server for a valid login. If user System-admin system will proceed to next process. System should grant access data from database.
Output :	Message of successful data enter, data save, and data retrieve from database.

Table 3.2 Function View, Add, Delete

Function: View and handle pending student registration

View and handle pending student registration is functional requirements for National Institute of Sports Science Student Management System. Table 3.3 depicts the details of this functional requirement.

Function :	View and handle pending student registration
Summary :	The system should allow the system admin view and modify details of pending student's details.
Input :	Pending student details

Process :	The system will check with the MSSQL server for a valid login. If the user is System-Admin. System will grant the access and modify the pending student details
Output :	Message of successful message alerts will display after changes.

Table 3.3 View and handle pending student registration

Function: Manage student records

Manage student records is functional requirements for National Institute of Sports Science Student Management System. Table 3.4 depicts the details of this functional requirement.

Function :	Mange Student Records
Summary :	The system should allow the System-Admin to successfully Modify details of Students.
Input :	Student’s details for required fields.
Process :	The system will check the login session of user. If system-admin is user system grant access the mange student details.
Output :	System will provide successful message alerts to user.

Table 3.4 Manage student records

Function: Sending e-mails to students and staffs.

Sending E-Mails to Students and Staffs is functional requirements for National Institute of Sports Science Student Management System. Table 3.5 depicts the details of this functional requirement.

Function :	Sending E-Mails to Students and Staffs
Summary :	The system should allow system administrator to send e-mails to students and staffs.
Input :	System admin login, and e-mail send details
Process :	The system will check with the MSSQL server for a valid login. If the username and password are valid and user is system-admin, the user can send e-mails

Output :	Message of successful alert display
----------	-------------------------------------

Table 3.5 Sending e-mails to students and staffs.

Function: Search, Delete, block, unblock, update or Assign user to different roles

Search, Delete, Block, Unblock, Update or Assign user to different roles are functional requirements for National Institute of Sports Science Student Management System. Table 3.6 depicts the details of this functional requirement.

Function :	Search, Delete, block, unblock, update or Assign user to different roles
Summary :	The system should allow the system admin to change privileges of users.
Input :	Username and password for system admin and selected users
Process :	The system will check for valid system admin login then process to selected user privileges change.
Output :	Message of successful display.

Table 3.6 Search, Delete, block, unblock, update or Assign user to different roles

Function: Send Newsletters to registered users

Send Newsletters to registered users is functional requirements for National Institute of Sports Science Student Management System. Table 3.7 depicts the details of this functional requirement.

Function :	Send Newsletters to registered users
Summary :	The system should allow system administrator to send e-mails with news to students and staffs.
Input :	System admin login, and e-mail send details

Process :	The system will check with the MSSQL server for a valid login. If the username and password are valid and user is system-admin, the user can send e-mails
Output :	Message of successful display

Table 3.7 Send Newsletters to registered users

Function: Handle Payment Gateway

Handle payment gateway is functional requirements for National Institute of Sports Science Student Management System. Table 3.8 depicts the details of this functional requirement.

Function :	Payment Gateway
Summary :	The system should allow the users to successfully login once the validated username and password are submitted and able to pay their courses using payment gateway.
Input :	User Login details, Electronic payment Card details
Process :	The system will check with the MSSQL server for a valid login. If the valid login proceeds to application programme interface of payment gateway.
Output :	Message of successful login message and message to mobile payment successful.

Table 3.8 Handle Payment Gateway

3.5.2 Functional Requirements User Group – Registered User

Registered user is functional requirements for National Institute of Sports Science Student Management System. Table 3.9 depicts the details of this functional requirement.

Function :	User Group – Register User
Summary :	System admin able to register different user types.

Input :	System admin login, Fill User Registration Form
Process :	The system will check with the MSSQL server for a valid login. If the provided username and password are valid, the system admin is capable to register users.
Output :	Message of successful login and display and message display for success register users.

Table 3.9 User Group – Register User

Function: Login with change and forgotten password

Change and forgotten password is functional requirements for National Institute of Sports Science Student Management System. Table 3.10 depicts the details of this functional requirement.

Function :	Login with change and forgotten password
Summary :	The system should allow the users to successfully login once the validated password to change. Forgotten password change their original password
Input :	Username, email and password
Process :	The system will check with the MSSQL server for a valid login. If valid login user can change their password also if user forgot password the user has to request email to password change one-time code link via Email.
Output :	Message of successful login and display and if change password successful message for change password

Table 3.10 Login with change and forgotten password

Function: View Schedule

View Schedule is functional requirements for National Institute of Sports Science Student Management System. Table 3.10 depicts the details of this functional requirement.

Function :	View Schedule
------------	---------------

Summary :	The system should allow the users to successfully login once according users roles display schedules
Input :	Username and password
Process :	The system will check with the MSSQL server for a valid login. If the username and password are valid, the main home page applicable to the user is displayed, else the user will be denied of access. User can select schedule navigation and view schedules.
Output :	Message of successful login and display of the home page else error message for invalid login.

Table 3.11 View Schedule

Function: View Courses

View is functional requirements for National Institute of Sports Science Student Management System. Table 3.12 depicts the details of this functional requirement.

Function :	View Courses
Summary :	The system should allow the users to successfully login once according users roles display courses
Input :	User name and password
Process :	The system will check with the MSSQL server for a valid login. If the user name and password are valid, the main home page applicable to the user is displayed, else the user will be denied of access. User can select schedule navigation and view courses.
Output :	Message of successful login and display of the home page else error message for invalid login.

Table 3.12 View Courses

Function: View Exam Details

View Exam Details is functional requirements for National Institute of Sports Science Student Management System. Table 3.13 depicts the details of this functional requirement.

Function :	View Exam Details
------------	-------------------

Summary :	The system should allow the users to successfully login once according users roles display exam details
Input :	Username and password
Process :	The system will check with the MSSQL server for a valid login. If the username and password are valid, the main home page applicable to the user is displayed, else the user will be denied of access. User can select schedule navigation and view exam.
Output :	Message of successful login and display of the home page else error message for invalid login.

Table 3.13 View Exam Details

Function: View Resource Details

View resource details is functional requirements for National Institute of Sports Science Student Management System. Table 3.14 depicts the details of this functional requirement.

Function :	View Resource Details
Summary :	The system should allow the users to successfully login once according users roles display resource details
Input :	Username and password
Process :	The system will check with the MSSQL server for a valid login. If the username and password are valid, the main home page applicable to the user is displayed, else the user will be denied of access. User can select schedule navigation and view resource.
Output :	Message of successful login and display of the home page else error message for invalid login.

Table 3.14 View Resource Details

3.6.2 Non-Functional Requirements

Security and safety of the system will be assured by using Authentication and Authorization. Each user will be authenticated by using unique username and password. Password will be saved by using hashing concept. Each user will be authorized by using Role based Security

mechanism. By that Administrator privileges and general user privileges can be implemented without any issues. Hence there will be mainly two Roles each with various permissions and privileges. Secure connections will be used in appropriate areas (SSL) so that data sent by users is encrypted. Credit card numbers has to be securely stored in the database. As the system is online and self-explanatory safety of the system is steady.

Maintainability

System is easy to maintain, update and modify due to the technology and it use.

Availability

System is fully available and operational for users 24/7 365 days.

Accuracy

Accuracy is up to the maximum level due to exquisite exception handling techniques which are offered with the ASP.Net framework.

Usability

System allows any users to login if the user is using HTML or any technology which has been derived from HTML. User can study by them self the system because of the user friendliness. Also no specific training required because most of the users are aware of fundamentals of web browsing and internet. System will work with all popular web browser such chrome, Firefox, Explore and etc.

Chapter 4 : Design/Methodology

4.1 Methodology

Software design is a process to transform user requirements into some suitable form, which helps the programmer in software coding and implementation.

4.1.1 Agile Methodology

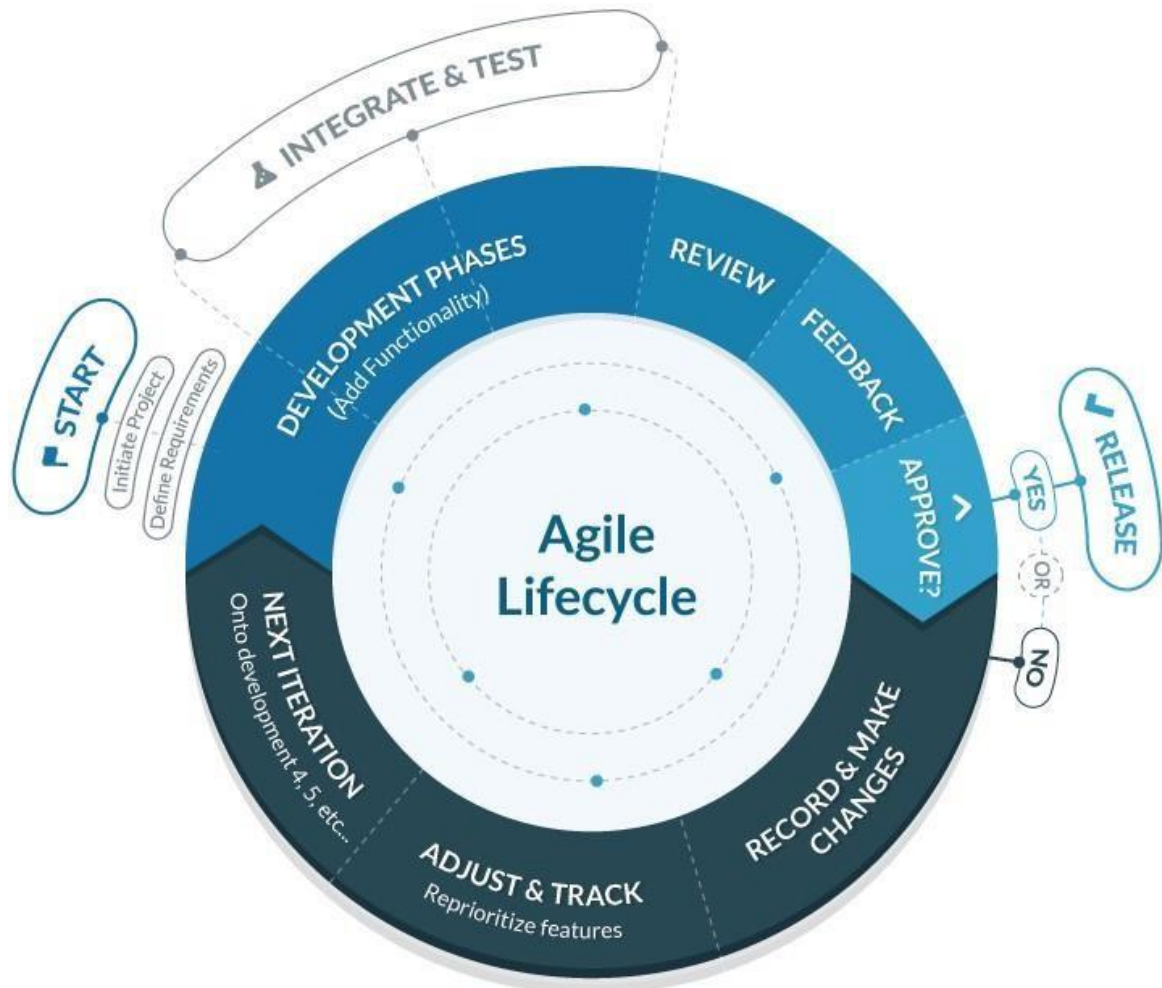


Figure 4.1.1.1 – agile model

The Agile methodology is an iterative methodology to project management. Instead of taking a “one-shot” approach, agile methodologies adapt to changing requirements.

Agile software development specifically, requires feedback from customers and cross functional teams for success. The agile method (figure 4.1.1.1) is a real-time approach to project management to help manage development as it occurs, in a changing manner. With the Agile methodology create, incorporate feedback into, test, and manage projects simultaneously.

According to market research conducted by the Project Management Institute, agile organizations fared substantially better than their non-agile counterparts in performance metrics.

For example:

- Seventy five percent of their goals compared to fifty six percent of non-agile organizations
- Sixty five percent of finished projects on time compared to the forty percent of non-agile organizations
- Sixty seven percent of finished projects on budget compared to Forty five percent of non-agile organizations
- Revenue grew thirty seven faster
- Agile organizations generated thirty percent more profits than non agile system

Clearly, there are benefits to using agile methods, in particular, can benefit from this project management approach.

4.1.2 Features of Agile Model

- Rapid and adaptive response to any change at any point in the project life cycle.
- Active communication among all stakeholders.
- Faster delivery of the software.
- Advantageous for short-lived projects like this project.
- Driven by customer requirements.

4.2 Design

Software design is a procedure to transform user requirements into some appropriate form, which helps the programmer in software coding and application.

For evaluating user requirements, an SRS (Software Requirement Specification) document is created whereas for coding and implementation, there is a need of more precise and detailed requirements in software terms. The output of this procedure can directly be used into implementation in programming languages.

Software design is the first step in SDLC (Software Design Life Cycle), which moves the attentiveness from problem domain to solution domain. It tries to specify how to fulfil the requirements mentioned in SRS

4.2.1 Architecture

Software application architecture is the procedure of describing an organized solution that meets all of the technical and operational requirements, while optimizing common quality characteristics such as performance, security, and manageability. It involves a sequence of decisions based on a wide range of factors, and each of these decisions can have considerable influence on the quality, performance, maintainability, and overall success of the application.

4.2.1.1 .NET Architecture

The Microsoft .NET architecture is the programming model for the .NET platform. The .NET Framework provides a managed execution environment, simplified development and deployment and integration with a wide variety of programming languages.

The .NET Framework has two key parts:

The .NET Framework class library is a comprehensive, object-oriented collection of reusable types that you can use to develop applications. The .NET Framework class library includes ADO.NET, ASP.NET, and Windows Forms.

The common language run-time (CLR) is the core run-time engine for executing applications in the .NET Framework. You can think of the CLR as a safe area - a "sandbox" - inside of which your

.NET code runs. Code that runs in the CLR is called managed code. It is fully protected from the outside environment and highly optimized within, taking advantage of the services that the CLR provides such as security, performance, deployment facilities, and memory management, including garbage collection.

4.2.1.2 MVC Architecture

Model View Controller or **MVC** as it is popularly called, is a software design pattern for developing web applications. A Model View Controller pattern is made up of the following three parts.

- **Model** – the lowest level of the pattern which is responsible for maintaining data.
- **View** – this is responsible for displaying all or a portion of the data to the user.
- **Controller** – Software Code that controls the interactions between the Model and View.

MVC is popular as it isolates the application logic from the user interface layer and supports separation of concerns. Here the Controller receives all requests for the application and then works with the Model to prepare any data needed by the View. The View then uses the data

prepared by the Controller to generate a final presentable response. The MVC abstraction can be graphically represented as follows figure 4.1.

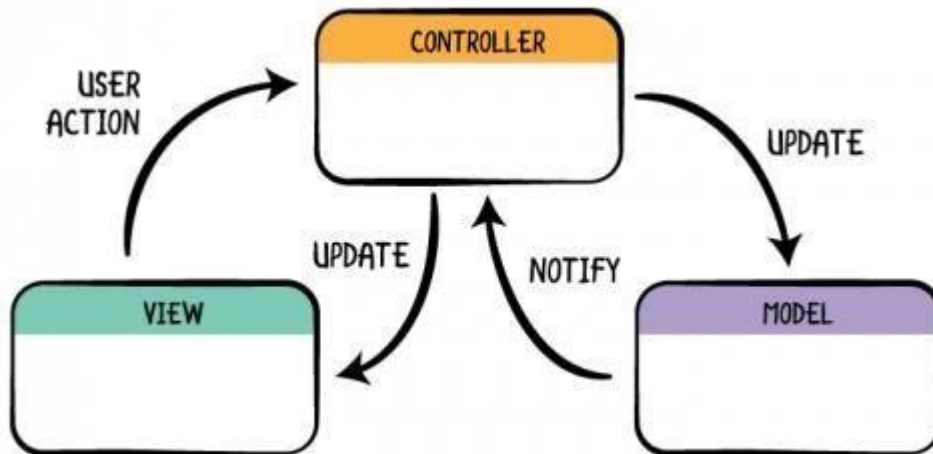


Table 4.1 MVC architecture

The Model

The model is responsible for managing the data of the application. It responds to the request from the view and it also responds to instructions from the controller to update itself.

The View

It means presentation of data in a particular format, triggered by a controller's decision to present the data. They are script-based template systems like JSP, ASP, PHP and very easy to integrate with AJAX technology.

The Controller

The controller is responsible for responding to the user input and perform interactions on the data model objects. The controller receives the input, it validates the input and then performs the business operation that modifies the state of the data model.

4.3 Design Pattern Repository Pattern

Design Patterns in object oriented world is reusable solution to common software design problems which occur again and again in real world application development. It is a template or description for how to solve a problem which can be used in many different situations.

The Repository Pattern is one of the most popular patterns to create an enterprise level application. It restricts us to work directly with the data in the application and creates new layers

for database operations, business logic and the application's UI. If an application does not follow the Repository Pattern, it may have the following problems:

- Duplicate database operations codes
- Need of UI to unit test database operations and business logic
- Need of External dependencies to unit test business logic
- Difficult to implement database caching, etc.

Using the Repository Pattern has many advantages:

- Business logic can be unit tested without data access logic;
- The database access code can be reused;
- Database access code is centrally managed so easy to implement any database access policies, like caching;
- It's easy to implement domain logics
- Domain entities or business entities are strongly typed with annotations; and more.

4.4 Use Case Identification

This Use case diagram referred to as behavior diagram of the Student Management Information System for NISS used to describe a set of actions that system can perform in collaboration with one or more external users of the system

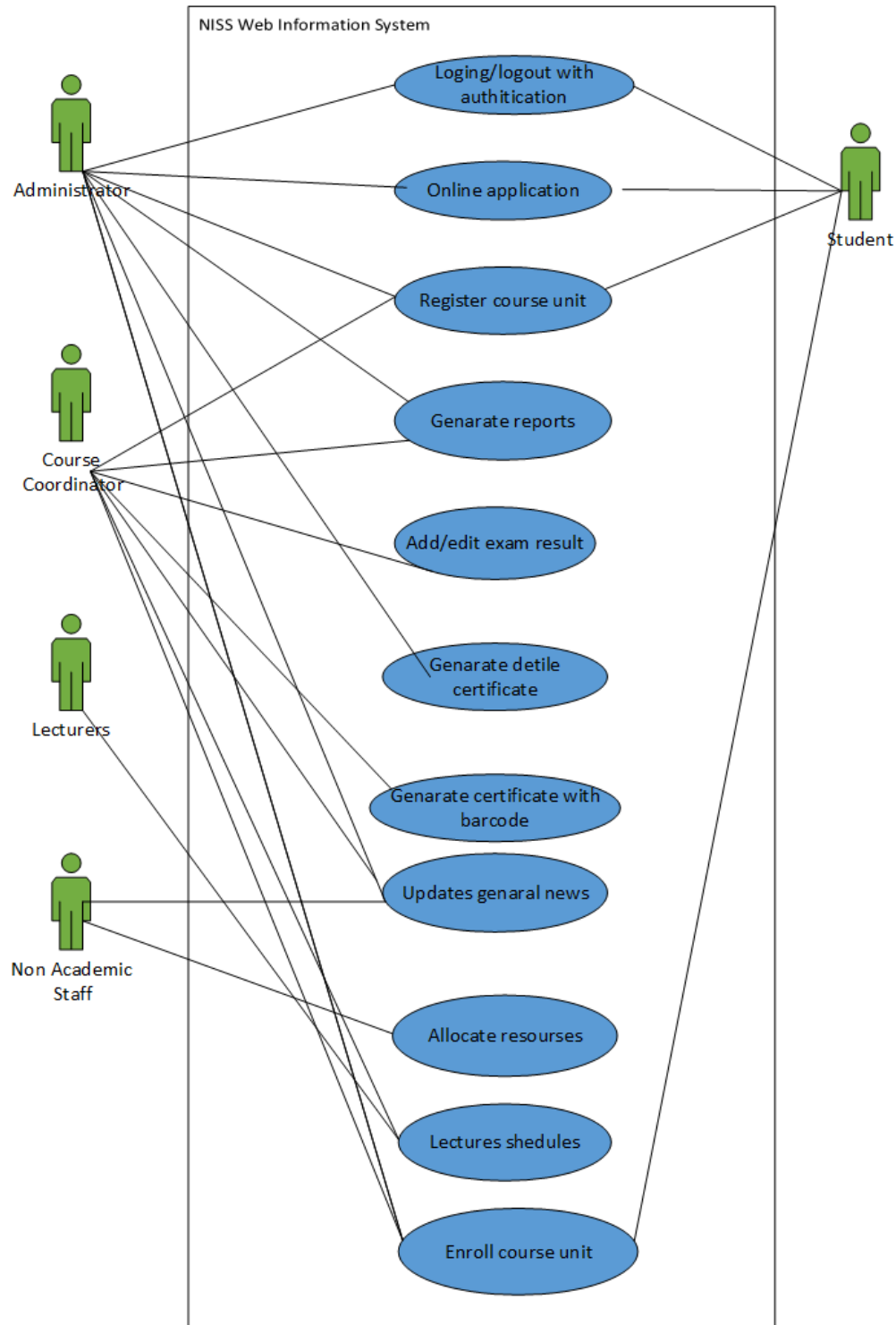


Figure 4.1 Use Case Diagram

4.5 Sequence Diagram

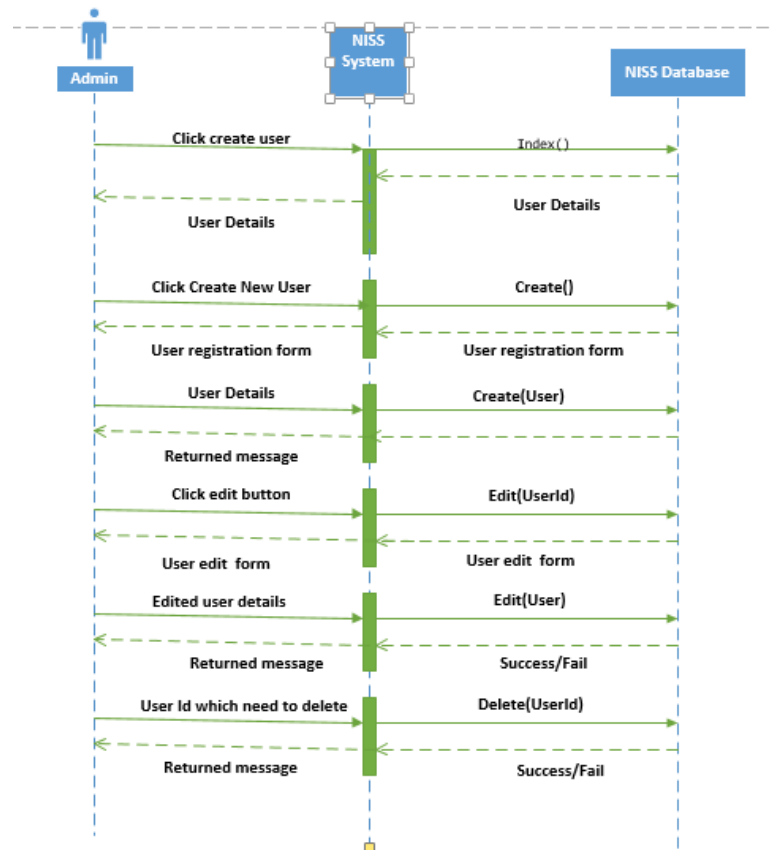


Figure 4.2 Sequence Diagram of admin-user function

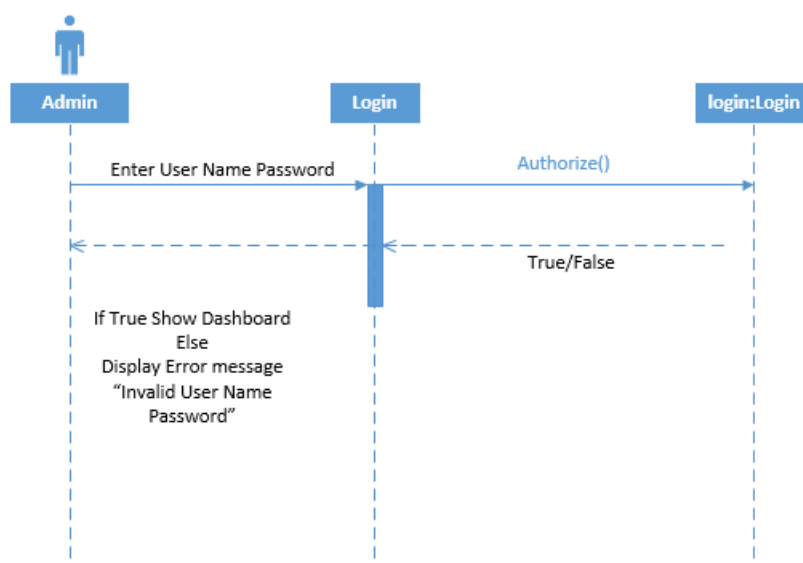


Figure 4.3 Sequence Diagram of administration login

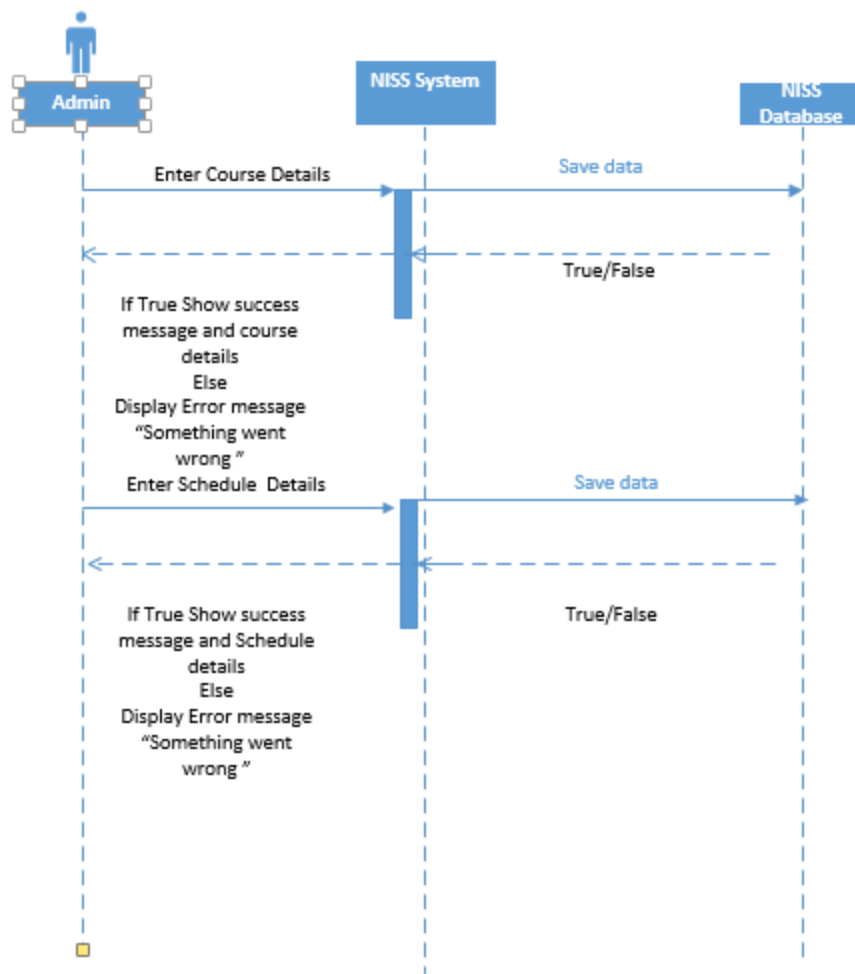


Figure 4.4 Sequence Diagram of admin- course details

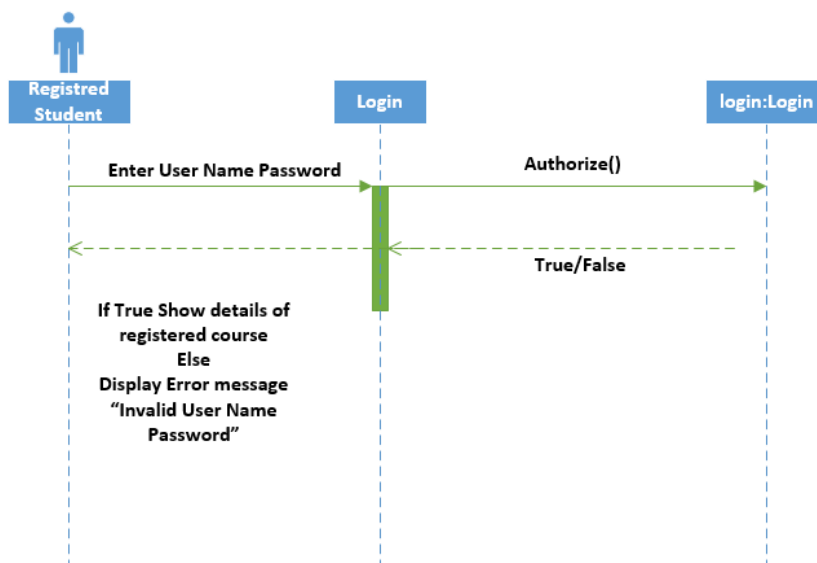


Figure 4.5 Sequence Diagram of registered student login

Chapter 5 : Implementation

5.1 Introduction

In development process, there are many Methods to develop a system. The system can be developed as described below.

Standalone – “A standalone computer system refers to any laptop or desktop computer that can run local applications on its own without needing a connection to a wide area network (WAN) or a local area network (LAN). All the application programs required for general use are installed on the hard disk”

Web based – “A Web-based application refers to any program that is accessed over a network connection using HTTP, rather than existing within a device’s memory. Web-based applications often run inside a Web browser. However, Web-based applications also may be client-based, where a small part of the program is downloaded to a user’s desktop, but the processing is done over the Internet on an external server”

In this developed system developer has used standalone method to develop the system. Reason to develop the system as a standalone system because this application will be only accessible for admin and trainers. Also, there are no many branches to network this application and use between branches. And also this system does not require an internet connection. Because of these reasons developers have been developing the system as Standalone system.

5.2 Good practices followed during the implementation

Due to time limitations or enthusiastic programmers who want instant outcomes for their code, commenting of code often takes a back seat. Code readability is a universal business in the world of computer programming. It's one of the first things we learn as developers. Therefore, when developing the MySQL query generator following good practices are followed:

- Commenting and documentation
- Consistent Indentation
- Avoid obvious comments
- Code grouping
- Consistent naming scheme
- Limit line length
- File and folder organization
- Separation Code and Data

According above listed good practices are followed in implementation process. For more readable code.

5.3 Implementation environment and Development tools

Software

The software we used to implement the system is Microsoft visual studio 2015. Language that is used to implement the system is C#. And as the database manager I used MSSQL 2014.

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs for Microsoft Windows, as well as web sites, web applications and web services. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can produce both native code and managed code.

Visual Studio includes a code editor supporting IntelliSense (the code completion component) as well as code refactoring. The integrated debugger works both as a source-level debugger and a machine-level debugger. Other built-in tools include a forms designer for building GUI applications, web designer, class designer, and database schema designer. It accepts plug-ins that enhance the functionality at almost every level including adding support for source-control systems (like Subversion) and adding new toolsets like editors and visual designers for domain specific or toolsets for other aspects of the software development lifecycle (like the Team Foundation Server client: Team Explorer).

SQL Server Management Studio (SSMS) is a software application first launched with the Microsoft SQL Server 2005 that is used for configuring, managing, and administering all components within Microsoft SQL Server. The tool includes both script editors and graphical tools which work with objects and features of the server

Entity Framework (EF) is an object-relational mapper that enables .NET developers to work with relational data using domain-specific objects. It eliminates the need for most of the data access code that developers usually need to write. Entity Framework allows you to create a model by writing code or using boxes and lines in the EF Designer. Both of these approaches can be used to target an existing database or create a new database.

Hardware

Hardware component used to implement the system

- Processor – i3 3.3GHZ 3rd gen

- RAM – 8 GB
- Monitor – ASUS 19 inches

5.4 Code and module structure description

I have used the pillar concept to implement the system

There are 4 pillar concepts in Object Oriented Programming.

1. Encapsulation
2. Abstraction
3. Inheritance
4. Polymorphism

Encapsulation

Encapsulation is the process of hiding an object's implementation from another object, while presenting only the interfaces that should be visible. Its inner workings are hidden to the client, which only invokes the interface methods.

Encapsulating a class

- Member of a class must always be declared with the minimum level of visibility.
- Provide setters and getters (also known as accessors / mutators) to allow controlled access to private data.
- Provide other public methods (known as interfaces) that other objects must adhere to in order to interact with the object

Setter are methods that (only) alter the state of an object. Getters are methods that return information about the state of an object.

Example:

- Encapsulation promotes maintenance
- Code changes can be made independently
- Increases usability
- Protects an object from unwanted access from the client

Following screen shot illustrate the way of use abstraction in National Institute of Sports Science implemented solution.

```
public int Id { get; set; }
public string ResourceCode { get; set; }
public string ResourceName { get; set; }
public int ScheduleId { get; set; }
public int ResourceTypeId { get; set; }
```

Figure 5.1 use of Encapsulation of project

Above screen shot illustrate the use of encapsulation concept during the project conducted.

Abstraction

Abstraction is the concept of taking some object from the real world, and converting it to programming terms. Such as creating a class and giving it Variables, properties, and methods.

Abstraction is "To represent the essential feature without representing the background details". This lets you focus on what the object does instead of how it does it. The abstraction provides you a generalized view of your classes or object by providing relevant information. Abstraction is the process of hiding the working style of an object, and showing the information about an object in an understandable manner.

Inheritance

Inheritance is a way by which a subclass inherits attributes and behaviours of a parent or base class and can have its own as well.

Inheritance allows child classes inherits the characteristics of Attributes, Operations parent class. A child class can extend the parent class to add new fields and methods, redefine methods. A class can implement an interface by providing implementation for all its methods.

Benefits of inheritance

- Extensibility
- Reusability
- Provides abstracts
- Eliminates redundant code
- Use inheritance for building is-a relationship

Following screen shot illustrate the way of use inheritance in National Institute of Sports Science implemented solution.

```
public partial class Student: User
{
    [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2214:DoNot
public Student()
    {
        this.Certificates = new HashSet<Certificate>();
        this.OverallGrades = new HashSet<OverallGrade>();
        this.Marks = new HashSet<Mark>();
    }

    public int Id { get; set; }
    public string StudentCode { get; set; }
    public string FullName { get; set; }
    public string Address { get; set; }
    public Nullable<int> AcademicYear { get; set; }
    public string Gender { get; set; }
    public string NIC { get; set; }
    public string MobileNo { get; set; }
    public string LandPhoneNo { get; set; }
    public string Password { get; set; }
    public string Email { get; set; }
    public string CivilStatus { get; set; }
    public int UserId { get; set; }
    public int StudentTypeId { get; set; }
    public string Occupation { get; set; }
}
```

Figure 5.2 Use of inheritance

Above screen shot illustrate the use of inheritance concept during the project conducted.

Polymorphism

Many forms of a single object are called Polymorphism.

Polymorphism refers to the ability to take into different forms or stages. A subclass can define its own unique behaviour and still share the same functionalities or behaviour of its Parent/base class. Subclass can have their own behaviour and share some behaviour from its parent class.

A parent class cannot have the behaviour of its subclass.

5.4.1 Barcode Generation

Important codes of NISS barcode generation function illustrated below.

```
public ActionResult Barcode()
{
    SqlConnection con = new SqlConnection("Data Source=.;Initial Catalog=Hospital;Integrated Security=True");
    string query = "select * From Product";
    DataTable dt = new DataTable();
    con.Open();
    SqlDataAdapter sda = new SqlDataAdapter(query, con);
    sda.Fill(dt);
    con.Close();
    IList<BarCodeModel> model = new List<BarCodeModel>();
    for (int i = 0; i < dt.Rows.Count; i++)
    {
        var p = dt.Rows[i]["BarCodeImage"];
        model.Add(new BarCodeModel()
        {
            Name = dt.Rows[i]["Name"].ToString(),
            Description = dt.Rows[i]["Description"].ToString(),
            Barcode = dt.Rows[i]["Barcode"].ToString(),
            ImageUrl = dt.Rows[i]["BarCodeImage"] != null ? "data:image/jpeg;base64," + Convert.ToBase64String((byte[])dt.Rows[i]["BarCodeImage"]) : ""
        });
    }
    return View(model);
}
```

Figure 5.3 Important codes of NISS barcode generation function

Sample Generated Barcode illustrated below



Figure 5.4 Sample Barcode

Chapter 6 : Testing and Evaluation

6.1 Importance of testing

Software Testing is necessary because we all make mistakes. Some of those mistakes are unimportant, but some of them are expensive or dangerous. Need to check everything and anything produce because things can always go wrong – humans make mistakes all the time. Software testing is very important because of the following reasons:

- Software testing is really required to point out the defects and errors that were made during the development phases.
- It's essential since it makes sure of the Customer's reliability and their satisfaction in
- The application.
- It is very important to ensure the Quality of the product. Quality product delivered to the customers helps in gaining their confidence. (Know more about Software Quality)
- Testing is necessary in order to provide the facilities to the customers like the delivery of high quality product or software application which requires lower maintenance cost and hence results into more accurate, consistent and reliable results.
- Testing is required for an effective performance of software application or product.
- It's important to ensure that the application should not result into any failures because it can be very expensive in the future or in the later stages of the development.
- It's required to stay in the business.

In that case, testing take part and partial place in software project. Therefore, conducting a testing according to proper test plan required. In that case implemented this Information System with barcode included certificate issue system and Support Android Mobile Application project testing properly.

6.2 Selected Testing Methods

To test this information System by following testing methods were selected. Because this IS codes need to test in wider context. Selected testing methods were:

- White Box Testing
- Black Box Testing
- Acceptance Testing

6.2.1 White Box Testing

White box testing is a way of testing the external functionality of the code by examining and testing the program code that realises the external functionality. This is also known as clear box, or glass box or open box testing.

6.2.2 Black Box Testing

Black box testing is testing strategy based solely on requirements and specification. Black box testing requires no knowledge of internal paths, structures, or implementation of the software being tested.

6.2.3 Acceptance Testing

When the development organization has performed its system test and has corrected all or most defects, the system will be delivered to the user or customer for acceptance testing. The acceptance test should answer questions such as Can the system be released?

6.3 Test Plan and Test Cases

6.3.1 Test Plan

Test plan is the project plan for the testing work to be done. It is not a test design specification, a collection of test cases or set of test procedures.

Following tables 6.1, 6.2, 6.3, 6.4 and 6.5 are depicts the conducted test plans for current project.

Test Plan for Student Management System for NISS	
Test Plan ID	Test001
Testing objectives	Current test plan is created for achieve following test objectives: <ul style="list-style-type: none">• Testing strategies to working• Identify the tool should test• To recognize test situations to execute the test process• Test the software deliverable currently listed
Testing Modules	Information Systems Login Module
Features planning to test	User connects to their login to the interface thorough information system of NISS. Using password and username (email) as inputs. After logged into to the system, need to test the functions are working properly.

Not Testing features	Speed of the system
Testing Method	Black Box And White Box Testing
Testing Environment	Window 10 Operating System, Visual Studio integrated Environment Version 2015, C# and ASP.NET
Testing Activities	Test Planning, Test Designing, Deploy Test, Conduct Test
Deliverables	Test Cases, Test Methods, Test Data
Agenda	2017-12-20 1.00 P.M 2018-01-10 2.00 P.M

Table 6.1 Test Plan for Student Management System for NISS 1

Test Plan for Student Management System for NISS	
Test Plan ID	Test002
Testing objectives	<p>Current test plan is created for achieve following test objectives:</p> <ul style="list-style-type: none"> • Testing strategies to working • Identify the tool should test • To recognize test situations to execute the test process • Test the software deliverable currently listed
Testing Modules	Insert, update, delete and search of admin section of IS - NISS
Features planning to test	Using NISS information system's data entry forms check whether successfully data saved and retrieved.
Not testing features	Speed of the system
Testing Method	Black Box And White Box Testing
Testing Environment	Window 10 Operating System, Visual Studio integrated Environment Version 2015, C# and ASP.NET
Testing Activities	Test Planning, Test Designing, Deploy Test, Conduct Test
Deliverables	Test Cases, Test Methods, Test Data
Agenda	2017-12-20 1.00 P.M 2018-01-10 2.00 P.M

Table 6.2 Test Plan for Student Management System for NISS 2

Test Plan for Student Management System for NISS	
Test Plan ID	Test003
Testing objectives	Current test plan is created for achieve following test objectives: <ul style="list-style-type: none"> • Testing strategies to working • Identify the tool should test • To recognize test situations to execute the test process • Test the software deliverable currently listed
Testing Modules	Insert, update, delete and search of Course Coordinator's section of IS -NISS
Features planning to test	Using NISS information system's data entry forms check whether successfully data saved and retrieved.
Not testing features	Speed of the system
Testing Method	Black Box And White Box Testing
Testing Environment	Window 10 Operating System, Visual Studio integrated Environment Version 2015, C# and ASP.NET
Testing Activities	Test Planning, Test Designing, Deploy Test, Conduct Test
Deliverables	Test Cases, Test Methods, Test Data
Agenda	2017-12-20 1.00 P.M 2018-01-10 2.00 P.M

Table 6.3 Test Plan for Student Management System for NISS 3

Test Plan for Student Management System for NISS	
Test Plan ID	Test004
Testing objectives	Current test plan is created for achieve following test objectives: <ul style="list-style-type: none"> • Testing strategies to working • Identify the tool should test • To recognize test situations to execute the test process • Test the software deliverable currently listed

Testing Modules	Insert, update, delete and search of Student's section of IS - NISS
Features planning to test	Using NISS information system's data entry forms check whether successfully data saved and retrieved.
Not Testing features	Speed of the system
Testing Method	Black Box And White Box Testing
Testing Environment	Window 10 Operating System, Visual Studio integrated Environment Version 2015, C# and ASP.NET
Testing Activities	Test Planning, Test Designing, Deploy Test, Conduct Test
Deliverables	Test Cases, Test Methods, Test Data
Agenda	2017-12-20 1.00 P.M 2018-01-10 2.00 P.M

Table 6.4 Test Plan for Student Management System for NISS 4

Test Plan for Student Management System for NISS	
Test Plan ID	Test005
Testing objectives	<p>Current test plan is created for achieve following test objectives:</p> <ul style="list-style-type: none"> • Testing strategies to working • Identify the tool should test • To recognize test situations to execute the test process • Test the software deliverable currently listed
Testing Modules	Insert, update, delete and search of Staff's section of IS - NISS
Features planning to test	Using NISS information system's data entry forms check whether successfully data saved and retrieved.
Not Testing features	Speed of the system
Testing Method	Black Box And White Box Testing
Testing Environment	Window 10 Operating System, Visual Studio integrated Environment Version 2015, C# and ASP.NET

Testing Activities	Test Planning, Test Designing, Deploy Test, Conduct Test
Deliverables	Test Cases, Test Methods, Test Data
Agenda	2017-12-20 1.00 P.M 2018-01-10 2.00 P.M

Table 6.5 Test Plan for Student Management System for NISS 5

6.3.2 Test Cases

Test Case Login

Login function is one of the most important feature in the National Institute Sports Science (NISS) Student Management System. Login function make sure level of access to the system and privacy of the details provided to the system. Therefore login function required test properly. In that case this test case login conducted.

Test Case #: Test001

Test Case Name: Information Systems Login Module

System: NISS Information System

Subsystem: Login

Designed by: P.H.A.U. De Silva

Design Date: 2017 – 12 – 20

Executed by: P.H.A.U. De Silva

Execution Date: 2018 – 01 – 10

Short Description: Test the login in to NISS Information System by using login interface

Preconditions: Required hosted web information system

Step	Input	Testing Method	Expected Output	Actual Output
01	Information System Server Login details, User name: password: Without username and password click login	Black Box	Successfully connect to the login of NISS information system. Display Validation message	Successfully connect to the login of NISS information system. Display Validation message Refer Figure 6.1 Below

02	Information System Server Login details, User name: password: Without username and password click login	Black Box	Successfully connect to the login of NISS information system. Display invalid credential message and Please enter a valid email address	Successfully connected to the login of NISS information system. Displayed Validation Message. Refer Figure 6.2 provided under appendix section
03	Information System Server Login details, User name: password:	White Box	Successfully connect to the login of NISS information system. Display invalid credential message and Please enter a valid email address	Successfully connected to the login of NISS information system. Displayed Validation Message. Refer Figure 6.3 provided under appendix section

Table 6.6 Test Case Login

Test Case – Admin Resource Type

Test Case #: Test002

Test Case Name: Insert, update, delete and search of admin section of IS -NISS

System: NISS Information System

Subsystem: Admin

Designed by: P.H.A.U. De Silva

Design Date: 2017 – 12 – 20

Executed by: P.H.A.U. De Silva

Execution Date: 2018 – 01 – 10

Short Description: Using NISS information system’s “Admin Resource Type” data entry forms check whether successfully data saved and retrieved.

Preconditions: Required hosted web information system and login of admin.

Step	Input	Testing Method	Expected Output	Actual Output
01	Without entering input click create button	Black Box	Display required validation Message	Display required validation Message Refer Figure 6.4 provided under appendix section
02	Without entering input click create button	White Box	Display required validation Message	Display required validation Message Refer Figure 6.5 provided under appendix section
03	Enter input and click create button	Black Box	Data successfully save and displaying on index page	Data successfully save and displaying on index page. Refer figure 6.6 provided under appendix section

04	Enter input and click create button	Black Box	Data successfully save and displaying on index page	Data successfully save and displaying on index page. Refer figure 6.6 provided under appendix section
05	Enter input and click create button	Black Box	Data successfully save and displaying on index page	Data successfully save and displaying on index page. Refer figure 6.7 provided under appendix section
06	Click edit button	Black Box	Data successfully edit and then save	Data successfully edit and then save . Refer figure 6.8 provided under appendix section
07	Click Delete button	Black Box	Data successfully delete from database by using delete button	Data successfully delete from database by using delete button. Refer figure 6.9 provided under appendix section
08	Click Details button	Black Box	Data can be retrieve according to the database	Data can be retrieve according to the database Refer figure 6.10 provided under appendix section

Table 6.7 Summary of testing 1

Test Case – Admin User Type

Test case user type is another key function of National Institute of Sports Science (NISS) Student Management System. This User Type Function allows to assign type for the user. User Type decides the which access level granted to the user therefore user type function testing is mandatory in that case Test Case – Admin User Type test case conducted.

Test Case #: Test003

Test Case Name: Insert, update, delete and search of admin section of IS -NISS

System: NISS Information System

Subsystem: Admin

Designed by: P.H.A.U. De Silva

Design Date: 2017 – 12 – 20

Executed by: P.H.A.U. De Silva

Execution Date: 2018 – 01 – 10

Short Description: Using NISS information system’s “Admin User Type” data entry forms check whether successfully data saved and retrieved.

Preconditions: Required hosted web information system and login of admin.

Step	Input	Testing Method	Expected Output	Actual Output
01	Without entering input click create button	Black Box	Display required validation Message	Display required validation Message Refer Figure 6.4 provided under appendix section
02	Without entering input click create button	White Box	Display required validation Message	Display required validation Message Refer Figure 6.5 provided under appendix section

03	Enter input and click create button	Black Box	Data successfully save and displaying on index page	Data successfully save and displaying on index page. Refer figure 5.6 provided under appendix section
04	Enter input and click create button	Black Box	Data successfully save and displaying on index page	Data successfully save and displaying on index page. Refer figure 6.6 provided under appendix section
05	Enter input and click create button	Black Box	Data successfully save and displaying on index page	Data successfully save and displaying on index page. Refer figure 6.9 provided under appendix section
06	Click edit button	Black Box	Data successfully edit and then save	Data successfully edit and then save Refer figure 6.10 provided under appendix section provided under appendix section
07	Click Delete button	Black Box	Data successfully delete from database by using delete button	Data successfully delete from database by using delete button. Refer figure 6.11 provided under appendix section
08	Click Details button	Black Box	Data can be retrieve according to the database	Data can be retrieve according to the database Refer figure 6.12 provided under appendix section

Table 6.8 Summery of Testing 2

6.3.3 Acceptance Testing

Acceptance testing is done by using selected beginner student of NISS in computing and Systems development. Sample Questionnaire Form Mentioned at Appendix Section -A Questionnaire form provided to give feedback of their experience when NISS information systems testing.

Questionnaire Form for User Feedback

Name of User: -

1.

	Strongly Agree	Agree	Somewhat Agree	Disagree	Strongly Agree
To be filled by those who are employed					
Do you think this system have all the features					
Do you think this system get any unexpected errors?					
Do you think this system can be implemented in your PC or Laptop more easily?					
Do you think is this system feels user friendly					

2. What are the drawbacks of this system?

.....

3. Is there any feature to be improved?

.....

4. How is the designing of this system?

Satisfied	Moderate	Need to be implement
-----------	----------	----------------------

5. Does this system work in Operating System?

Yes		No	
-----	--	----	--

6. Do all the features work without any problem?

.....
.....

7. What are the other important things in this system?

.....
.....

8. Comments

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

.....

Signature of User

6.3.3.1 Acceptance testing Questionnaire Analysis

The section 6.3. Illustrated questionnaire form provide to 10 novices of NISS staff. Most of them did not had knowledge about current system. There for the collected the feedbacks and analysis.

6.3.3.2 Hypotheses

Developed the following hypothesis in order to assess the relationship between the generated system and users of the NISS

H1. - There is a positive relationship between Information system and users NISS

H0 – There is a no relationship between information system of NISS and users

6.3.3.2.1 Analysis of Data Using Likert Scale

According to the average value, the responses of the employees can be used to test the hypotheses.

The questionnaire contains 08 questions and the question number 01 contains 4 sub questions which could be analysed through the Likert Scale. Each of those four questions has five options as shown below.

Option	Value Given
Strongly Agree	2
Agree	1
Neither Agree nor Disagree	0
Disagree	-1
Strongly Disagree	-2

Table 6.9 Weighted Value 1

As shown above the questions are weighted by giving +2 as the highest value for the most favourable answer and -2 as the lowest value for most unfavourable answer.

Calculation of the Average Value Using the Likert Scale

	Strongly Agree	Agree	Somewhat Agree	Disagree	Strongly disagree
To be filled by those who are employed					
Do you think this system have all the features	8	1	1	0	0
Do you think this system not get any unexpected errors?	7	1	1	1	0
Do you think this system can be implemented in your PC or Laptop more easily?	9	0	1	0	0
Do you think is this system feels user friendly	6	2	1	1	0

Table 6.10 Responses for the Likert Scale

As shown in the above table the total number of responses each category received must be put into a table first. Then each response must be weighted. For example, the question number 1 has obtained 11 responses for the Strongly Agree option. The value given for a Strongly Agree response is +2. Therefore, in order to get the value for the strongly agree category of the question number 1 the number of responses must be multiplied by the value given.

Strongly agree responses $8 \times 2 = 16$ weighted value

Likewise, all the responses received for each category of the question number one must be multiplied by the given value and then all the five values gained must be added together to get the total marks received for the question number one. Then, the mean value should be found out by dividing the total sum by the number of responses received.

	Strongly Agree	Agree	Neither Agree Nor Disagree	Disagree	Strongly Disagree	Total Sum	Average
Q 1	8 X 2	1 X 1	1 X 0	0 X -1	0 X -2	17	17/10
Q 2	7 X 2	1 X 1	1 X 0	1X -1	0 X -2	14	14/10
Q 3	9 X 2	0X 1	1X 0	0X -1	0X -2	18	18/10
Q 4	6 X 2	2 X 1	1 X 0	1 X -1	0 X -2	13	13/10
Total Average							6.2
Mean Value							1.55

Table 6.11 Mean Value Calculation

As shown in the above table the mean value of all the questions must be calculated and then the total average can be calculated by adding up the four average values. Then the total mean value should be divided by the number of questions in order to obtain the mean value for the targeted question, in this case for the question number 01 of the questionnaire.

The mean value obtained here is a positive value and moreover it is even closer to the value given to the most favourable answer; which is + 2. Therefore, it is denoted there is positive relationship with information system and users of NISS.

6.3.3.2.2 Analysis of the Designing of this system

Satisfied	Moderate	Need to be implement
7	2	1

Table 6.12 Analysis of system design

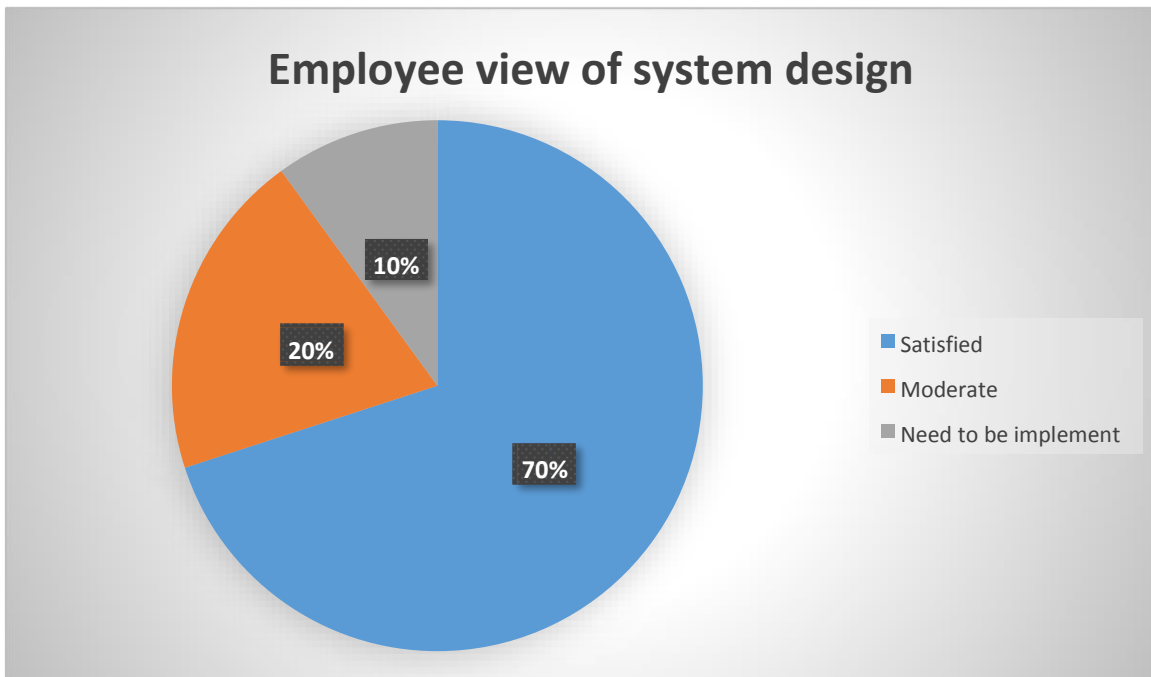


Figure 6.1 Employee Suggestion of System Design

The above graph shows the employee suggestion of system design of NISS. According to the graph we can see that 70 percent of employees satisfied for design of the system. Other 20 percent of employees moderately satisfied and remain 10 percent says system need to improve well.

6.3.3.2.3 Analysis of the system work in Operating System

NISS users has no of operating system of their desktop and laptops. Windows 7, windows 8, windows 8.1 and windows 10 are they used in NISS. In question number 5 asked “system work in operating system properly”

All participated was replayed as yes. Therefor can be decided the system was highly compatible for operating system.

According to above conclusion, most of users are satisfy the functionality of the system and ask for the expand features wider range. Therefore, this NISS information system provide overall satisfaction to the users.

Chapter 7 : Conclusion

7.1 Introduction

In conclusion, this Information System with barcode included certificate issue system and Supported Android Mobile Application (NISS IS) project was successfully implemented after following various stages in software development life cycle. This project was a development based documented module assessment and during this project I identified requirement and specification for NISS IS project.

NISS is an only one government academic Institute of Sports sector in Sri Lanka. Therefore when doing requirement gathering stage, identified special academic activities in NISS. I was identified there have various platforms and used various type of users in this system. In that case the selected scope of this project to develop the limited number of activities of NISS, specially selected academic works in NISS. To select the limit of development I received advices from project supervisor who recommended to select proper activity categories to implement within the time period given. Then uses development documents and literatures done for this NISS IS project. Using those document, the selected the scope of only academic works in NISS and got approved by supervisor before implement. After scope is clearly identified, functional and non-functional requirement identified using existing system analysing. During the implementation some modules were took time more than estimated time therefore required to work more time than expected.

In general, 99 percent of declared requirement objectives were met and users were provided positive feedbacks to this research based system tool. But there will be few suggestions in the future to develop this tool those recommendations mentioned in section 6.2. According to user feedback and supervisor advices NISS IS development was successfully completed.

7.2 Lessons Learned During the Project

Throughout, this project learned lot of important factors. Most important factor was the time management during conducting of project. Scheduling the project is not always accurate some tasks were taking little longer time than scheduled time. Then had to manage other task time according to task. Next important factor is self-learning and self-problem solving during the project also learned lot of programming knowledge, use of proper architectural and methodologies.

Also knowledge of framework is very important. Therefore, conduct this project has to learn .NET framework selected the ASP.NET and learned the required knowledge for conduct this project. Tools are very important learning about tools make easier to conduct this project. For diagram Microsoft Visio, and as Integrated Development Environment Visual Studio 2015 had to learn during this project. Visual Studio is making development easier. Learning about Visual Studio became very useful.

Other important thing is how to identify scope of the project. Then solutions generation is more complex and the use of theories is very difficult. The main thing understand from this project is there is no a method for system development. Development should customize according to the requirement of the system.

In conclusion, during this project lot of lessons are learned such as project management, programing languages, web technologies, tools and framework etc. Every learned thing makes project requirement achievable therefore learned lessons are very important to conduct this project and similar project like this.

Aims	Achieved	Concentration level
1) To help new students, existing students, in-house and visiting lectures of the NISS can access ISNISS	yes	High
2) To maintain and develop links students for further study.	yes	High
3) To assist in the provision and dissemination of information about the destinations of those who have completed their studies at National institute of Sports Science, Government and other interested parties.	yes	High
4) To inform education, information and guidance within the NISS.	yes	High
5) To collaborate with the Institute, academic departments, Ministries for shearing information	yes	High
6) To recruit, train and develop staff at all levels, by using educational portal by online.	yes	High
7) Integrate and support new learning and teaching opportunities and technologies for students and staff.	yes	High
8) Enable enrolment for students to perform basic administrative functions and tasks after selection. (ex: select specific subject for own stream)	yes	High
9) Support open interfaces and integration with applications and database systems	yes	High
10) Ensure data integrity, privacy, and security in an open access environment	yes	High
11) To source and provide up-to-date information about courses, fields of work.	yes	High
12). To promote the exchange of ideas between members of the NISS	yes	High

Table 7.1 Successfully rate of the project

7.3 Critical Assessment of the project

Implemented NISS information system is achieved the required scope. But comparing to the existing systems available around the world there is more improvement can be done to the implemented NISS information system. But within this time period more improvement or modification is not possible.

Existing systems are well time taken developed systems. Also few versions are already released therefore implemented system can little behind the existing systems detailed under section 2.1. Following table illustrate the comparison of existing systems and implement system.

Feature	NISS Information System	OpenSIS	Fedena	Gibbon
User Friendliness	High	High	Moderate	High
Maintainability	High (Developed for Specifically NISS)	Moderate for (Developed Generic Purpose)	Moderate for (Developed Generic Purpose)	Moderate for (Developed Generic Purpose)
Search ability	High	High	High	High
Efficiency	High	High	High	High
Cost	Free	Licenced	Licenced	Licenced
Security	High	High	High	High

Table 7.2 Critical Assessment of the Project

According above table this NISS system is the best suit for implemented for NISS. Existing other systems has problem over maintaining and cost. But NISS implemented system is designed and developed specifically for NISS. Therefore, this system can improve and best suits for the NISS as their information system.

7.4 Limitations and Future Recommendation

Limitations:

- Security enhancement
- Some same feathers in multiple database
- Customization required for few functions

Future Recommendation

There are a number of future directions would be valuable for National Institute of Sports Science implemented system can developed. Following recommendations are for developed NISS Student management make more efficient.

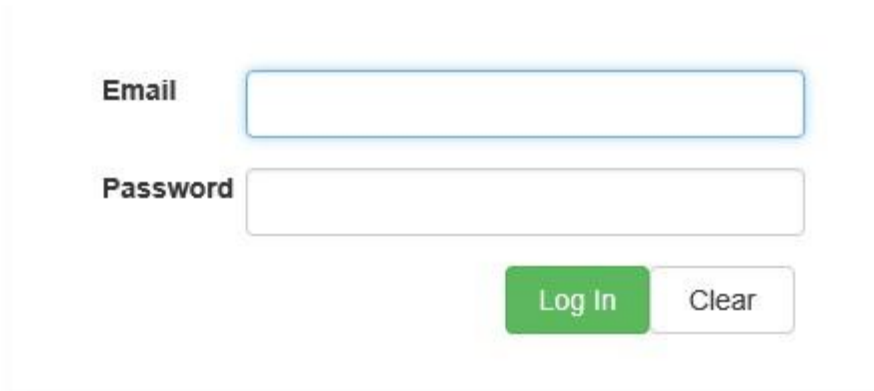
- Develop for light version for mobiles
- Other than using visual studio application interface with MVC prefer using Angular JS interface for more attractive User Interface.
- Cover all the functions of NISS
- Add feature one time multi user working
- Add analysing tools for this tool will provide more value

Chapter 8 : References

- [1] L. Hawryszkiewicz, System Analysis and Designing, 4th Edition, 2000.
- [2] W. V. ,Microsoft SQLServer 2000 Database Design and Implementation, 2nd edition, Microsoft Press, 2002..
- [3] blog.capterra.com, “5-ways-to-teach-agile-methodology-to-your-tech-team,” [Online]. Available: <https://blog.capterra.com>. [Accessed 17 October 2017].
- [4] “What is the microsoft .Net Architecture,” [Online]. Available: <https://www.progress.com/faqs/datadirecthttps>. [Accessed 20 October 2017].

Appendix A – User Documents

Log In



Email

Password

Figure 8.1 Login Form

Log In

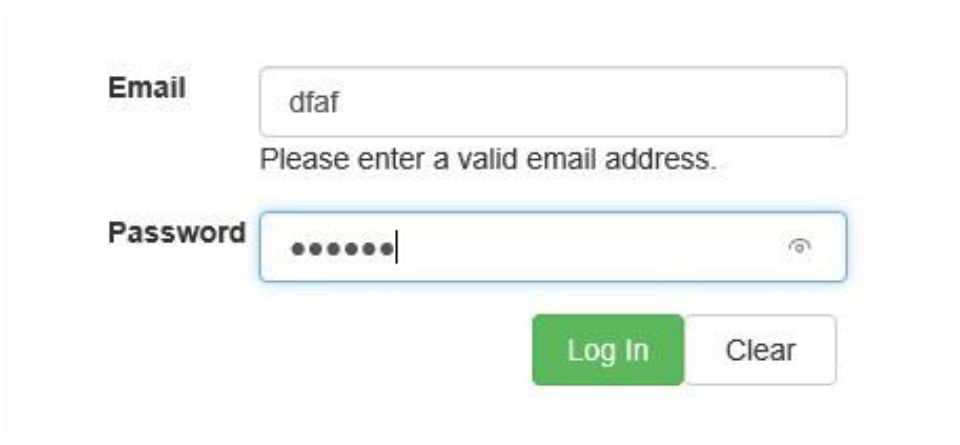


Email |
This field is required

Password
This field is required

Figure 8.2 Login Form with Validation Messages

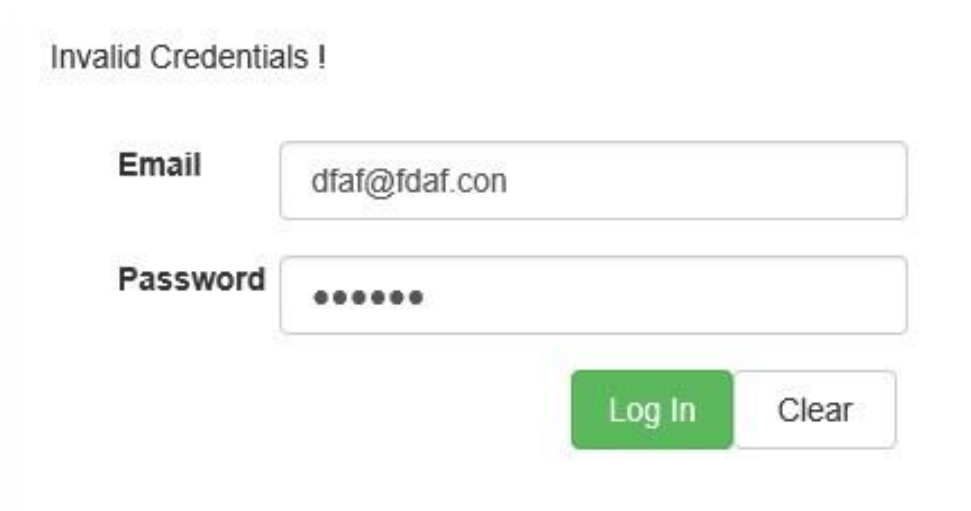
Log In



The login form contains two input fields. The 'Email' field has the text 'dfaf' and a red error message below it: 'Please enter a valid email address.'. The 'Password' field contains seven dots and a toggle icon. Below the fields are two buttons: a green 'Log In' button and a white 'Clear' button.

Figure 8.3 Login Form Enter Details

Log In



The login form displays an error message at the top: 'Invalid Credentials !'. The 'Email' field contains 'dfaf@fdaf.con' and the 'Password' field contains seven dots. The 'Log In' and 'Clear' buttons are visible at the bottom.

Figure 8.4 Invalid Credential Message

Create

ResourceType

Resource Type Name

The Resource Type Name field is required.

Create

[Back to List](#)

© 2018 - My ASP.NET Application

Figure 8.5 Screen Shot of Create Form Resource Type

```
namespace NISS_SMS.Models
{
    using System;
    using System.Collections.Generic;
    using System.ComponentModel.DataAnnotations;

    public partial class ResourceType
    {
        [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2214:DoNotCallOverridableMethodsInConstructors")]
        public ResourceType()
        {
            this.Resources = new HashSet<Resource>();
        }

        public int Id { get; set; }
        [Display(Name = "Resource Type Name")] [Required]
        public string ResourceTypeName { get; set; }

        [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2227:CollectionPropertiesShouldBeReadOnly")]
        public virtual ICollection<Resource> Resources { get; set; }
    }
}
```

Figure 8.6 Screen shot of Resource Type Model Class

Index

Create New

Resource Type Name	
Class Room	Edit Details Delete

© 2018 - My ASP.NET Application

Figure 8.7 Resource Type Index Page

```
// GET: ResourceTypes
public ActionResult Index()
{
    return View(db.ResourceTypes.ToList());
}
```

Figure 8.8 Screen shot of Resource Type Index Controller

```
// GET: ResourceTypes/Create
public ActionResult Create()
{
    return View();
}

// POST: ResourceTypes/Create
// To protect from overposting attacks, please enable the specific properties you want to bind to, for
// more details see http://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Create([Bind(Include = "Id,ResourceTypeName")] ResourceType resourceType)
{
    if (ModelState.IsValid)
    {
        db.ResourceTypes.Add(resourceType);
        db.SaveChanges();
        return RedirectToAction("Index");
    }

    return View(resourceType);
}
```

Figure 8.9 Screen Shot of Resource Type create controller Method

Index

[Create New](#)

Resource Type Name	
Class Room	Edit Details Delete
Exam Room	Edit Details Delete
Laptops	Edit Details Delete
Multimedia Projector	Edit Details Delete
Desktops	Edit Details Delete

© 2018 - My ASP.NET Application

Figure 8.10 Screenshot of successfully resource type record added

Edit

ResourceType

Resource Type Name

[Back to List](#)

© 2018 - My ASP.NET Application

Figure 8.11 Screenshot of editing records

Delete

Are you sure you want to delete this?

ResourceType

Resource Type Name Desktop Computers

| [Back to List](#)

© 2018 - My ASP.NET Application

Figure 8.12 Screenshot of delete view resource type

Details

ResourceType

Resource Type Name Laptops

[Edit](#) | [Back to List](#)

© 2018 - My ASP.NET Application

Figure 8.13 Screenshot of Detail View

Create

Staff

StaffCode	<input type="text"/>
FullName	<input type="text"/>
Email	<input type="text"/>
Password	<input type="password"/>
Gender	<input type="text"/>
MobileNo	<input type="text"/>
LandPhoneNo	<input type="text"/>
Occupation	<input type="text"/>
HPQ	<input type="text"/>
Religion	<input type="text"/>
ProfilePicture	<input type="text"/>
City	<input type="text"/>
HEQ	<input type="text"/>
DOB	<input type="text"/>
OtherInfo	<input type="text"/>
	<input type="button" value="Create"/>

[Back to List](#)

Lecturer

LecturerCode	<input type="text" value="L001"/>
FullName	<input type="text" value="Anita De Silva"/>
Gender	<input type="text" value="Female"/>
MobileNo	<input type="text" value="011*****"/>
LandPhoneNo	<input type="text" value="011*****"/>
Occupation	<input type="text" value="IT Lecturer"/>
Header	<input type="text" value="L1000"/>
OtherInfo	<input type="text"/>
Address	<input type="text" value="No 45/1 Pannipitya"/>
NIC	<input type="text" value="98*****99"/>
CivilStatus	<input type="text" value="Married"/>
DOB	<input type="text" value="1963/04/15"/>
HEQ	<input type="text" value="Bachelor of Science"/>
HPQ	<input type="text" value="-"/>
Religion	<input type="text" value="Buddhist"/>
City	<input type="text" value="Pannipitya"/>
ProfilePicture	<input type="text" value="-"/>
	<input type="button" value="Create"/>

[Back to List](#)

Figure 8.14 Form Staff Users

```
// GET: Staffs/Create
public ActionResult Create()
{
    ViewBag.UserId = new SelectList(db.UserTypes, "Id", "UserType1");
    return View();
}

// POST: Staffs/Create
// To protect from overposting attacks, please enable the specific properties you want to bind to, for
// more details see http://go.microsoft.com/fwlink/?linkid=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Create([Bind(Include = "Id,StaffCode,FullName,Gender,MobileNo,LandPhoneNo,Occupation,"
+ "HPQ,Religion,ProfilePicture,City,HEQ,DOB,UserId,OtherInfo")] Staff staff, User user)
{
    if (ModelState.IsValid)
    {
        var UserCode = Convert.ToInt32(db.Users.Max(m => m.Id));
        user.UserId = 2;
        user.UserId = "U" + UserCode;

        db.Users.Add(user);
        db.SaveChanges();

        var UserID = db.Users.Where(u => u.Email == user.Email).Select(s => s.Id).SingleOrDefault();

        staff.UserId = UserID;

        db.Staffs.Add(staff);
        db.SaveChanges();
        return RedirectToAction("Index");
    }

    ViewBag.UserId = new SelectList(db.UserTypes, "Id", "UserType1", staff.User.UserId);
    return View(staff);
}
```

Figure 8.15 Screen shot of Create Staff Controller Method

Index

[Create New](#)

StaffCode	FullName	Gender	MobileNo	LandPhoneNo	Occupation	HPQ	Religion	ProfilePicture	City	HEQ	DOB	OtherInfo	UserCode
ST002	Amila De Silva	Female	071*****	0112*****	IT Admin	-	Buddhist		-	Bachelor of Science	3/15/1982 12:00:00 AM	-	U1006

Figure 8.16 Screen Shot of Staff Index

```
public class StaffsController : Controller
{
    private NISS_SMSEntities db = new NISS_SMSEntities();

    // GET: Staffs
    public ActionResult Index()
    {
        var staffs = db.Staffs.Include(s => s.User);
        return View(staffs.ToList());
    }
}
```

Figure 8.17 Screen Shot of Staff Index Controller

Create

Staff

StaffCode	<input type="text" value="ST002"/>
FullName	<input type="text" value="Amila De Silva"/>
Email	<input type="text" value="amila@gmail.com"/>
Password	<input type="password" value="*****"/>
Gender	<input type="text" value="Female"/>
MobileNo	<input type="text" value="071*****"/>
LandPhoneNo	<input type="text" value="0112*****"/>
Occupation	<input type="text" value="IT Admin"/>
HPQ	<input type="text" value="-"/>
Religion	<input type="text" value="Buddhist"/>
ProfilePicture	<input type="text"/>
City	<input type="text" value="-"/>
HEQ	<input type="text" value="Bachelor of Science"/>
DOB	<input type="text" value="1982/03/15"/>
OtherInfo	<input type="text" value="-"/>
	<input type="button" value="Create"/>

[Back to List](#)

Figure 8.18 Staff Create form with Inputs

```

public ActionResult Edit(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Staff staff = db.Staffs.Find(id);
    if (staff == null)
    {
        return HttpNotFound();
    }
    ViewBag.UserId = new SelectList(db.Users, "Id", "UserCode", staff.UserId);
    return View(staff);
}

// POST: Staffs/Edit/5
// To protect from overposting attacks, please enable the specific properties you want to bind to, for
// more details see http://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Edit([Bind(Include = "Id,StaffCode,FullName,Gender,MobileNo,LandPhoneNo,Occupation,HPQ,
{
    if (ModelState.IsValid)
    {
        db.Entry(staff).State = EntityState.Modified;
        db.SaveChanges();
        return RedirectToAction("Index");
    }
    ViewBag.UserId = new SelectList(db.Users, "Id", "UserCode", staff.UserId);
    return View(staff);
}
}

```

Figure 8.19 - Screenshot of Staff Edit Controller Method

Delete

Are you sure you want to delete this?

Staff

StaffCode	ST002
FullName	Amila De Silva
Gender	Female
MobileNo	071*****
LandPhoneNo	0112*****
Occupation	IT Admin
HPQ	-
Religion	Buddhist
ProfilePicture	-
City	-
HEQ	Bachelor of Science
DOB	3/15/1982 12:00:00 AM
OtherInfo	-
UserCode	U1006

| [Back to List](#)

Figure 8.20 - Screenshot of Delete View Staff

Appendix B – Reports

Monthly Attendance Report

ID	Total Attendance	Attendance %
U0008	1	4.16666666666667 %
U0001	3	12.5 %

Certificate Details

Student ID	Full Name	Course Name	Certificate Code	Barcode
S001	Chanaka Madusanka	My Course	001	

No Of Students

Course ID	Course Name	Number of Registered Students
CC0001	My Course	1
CC0003	My Course 3	1
CC0002	My Course 2	2

Personal Details Report

User Type

ID	Full Name	Address	Date of Birth	Email	Telephone	Remarks
L001	Amila De Silva	No 45/1 Pannipitiya	3/15/1982 12:00:00 AM	lec@gmail.com	0112*****	-
L0002	Chanaka Madusanka	49/1 Cardinala Cooray Mawatha, Hendala,	7/25/1992 12:00:00 AM	chanaka@email.com	0112934033	-
L003	Madushan Sasanka	-	12/29/1996 12:00:00 AM	S@gmail.com	-	-
ST0001	Staff 1		7/25/1992 12:00:00 AM	chanaka@email.com	0112934033	-
S001	Chanaka Madusanka	49/1 Cardinal Cooray Mawatha, Hendala, Wattala	7/25/1992 12:00:00 AM	chana@gmail.com	0112934033	-
S0001	Test Visitor	Test Visitor	7/25/1992 12:00:00 AM	Test Visitor@	Test Visitor	Test Visitor
S0002	Test Visitor	Test Visitor	7/25/1992 12:00:00 AM	Test Visitor@	Test Visitor	Test Visitor
S0003	K.G.C.Madusanka	49/1 Cardinala Cooray Mawatha, Hendala,	7/25/1992 12:00:00 AM	S@gmail.com	0719948416	-

Figure 8.21 Screen shot of Personal Details Report, No of Students, Monthly Attendance

Course Coordinator Details Report

Resource

My Course

Confirm

Course Code	Course Name	Full Name	Highest Education Qualifications	Highest Professional Qualifications
CC0001	My Course	Amila De Silva	Bachelor of Science	-
CC0002	My Course 2	Amila De Silva	Bachelor of Science	-
CC0003	My Course 3	Amila De Silva	Bachelor of Science	-

Resource Allocation Report

Resource

Lecture Room B1

Confirm

Resource Name	Schedule	Schedule Start Time	Schedule End Time
Lecture Room B1	Schedule 1	12/12/2018 12:00:00 AM	12/12/2019 12:00:00 AM

Course Schedule Report

Start Time

mm/dd/yyyy

End Time

mm/dd/yyyy

Submit

Schedule Name	Course Name	Schedule Start Time	Schedule End Time
Schedule 1	My Course	12/12/2018 12:00:00 AM	12/12/2019 12:00:00 AM
Schedule 1	My Course 2	12/12/2018 12:00:00 AM	12/12/2019 12:00:00 AM
Schedule 1	My Course 3	12/12/2018 12:00:00 AM	12/12/2019 12:00:00 AM

Figure 8.22 Screen shot of Course Schedule Report, Resource Allocation

User Education Qualification

User Type

Confirm

User Type	Full Name	Highest Education Qualifications
Lecturers	Chanaka Madusanka	B.Eng Software Engineering
Lecturers	Test Visitor	Test Visitor
Lecturers	Test Visitor	Test Visitor
Lecturers	K.G.C.Madusanka	Bsc
Admin	Amila De Silva	Bachelor of Science
Admin	Chanaka Madusanka	-
Lecturers	Madushan Sasanka	-
Admin	Staff 1	Bsc

Gender Wise User Report

User Type

Confirm

User ID	Full Name	Gender
U0009	Chanaka Madusanka	Male
U0005	Test Visitor	Male
U0007	Test Visitor	Male
U0008	K.G.C.Madusanka	Male
U0008	Amila De Silva	Female
U0001	Chanaka Madusanka	Male
U0008	Madushan Sasanka	Male
U0001	Staff 1	Male

Figure 8.23 Screen shot of Gender wise user report

User Type Report

User Type

User ID	Full Name	User Type
U0009	Chanaka Madusanka	B.Eng Software Engineering
U0005	Test Visitor	Test Visitor
U0007	Test Visitor	Test Visitor
U0008	K.G.C.Madusanka	Bsc
U0008	Amila De Silva	Bachelor of Science
U0001	Chanaka Madusanka	-
U0008	Madushan Sasanka	-
U0001	Staff 1	Bsc

Figure 8.24 Screen shot of User Type Report

Appendix C – Test Cases

Test Case – Staff

Test case – Staff conducted for check that all the features of adding records, updating dating records, receiving records and deleting records are check whether properly working. This test case is important to staff users.

Test Case #: Test001

Test Case Name: Staff

System: NISS Information System

Subsystem - Staff

Designed by: Amila De Silva

Design Date: 2018 – 01 – 10

Executed by: Amila De Silva

Execution Date: 2018 – 01 – 10

Short Description: Testing the staff section insert, edit, details and delete

Preconditions: Authenticated staff login

Step	Input	Testing Method	Expected Output	Actual Output
01	Entering required to details to create form of staff section and click create button	Black Box	Successfully saving and display on index page.	Successfully saved and display on index page.

02	Entering required to details to create form of staff section and click create button	White Box	Successfully saving and display on index page.	Successfully saved and display on index page.
03	Check Index page saved Data display	Black Box	Successfully Displaying data	Successfully Displaying data Refer Figure 5.2 provided under appendix section
02	Check Index page saved Data display	White Box	Successfully Displaying data	Successfully Displaying data Refer Figure 5.2 provided under appendix section
03	Check Edit Page Edit Data Save	Black Box	Change data should successfully save and display on index	Change data should successfully save and displayed on index
03	Check Edit Page Edit Data Save	White Box	Change data should successfully save and display on index	Change data should successfully save and displayed on index
05	Test Delete Page	Black Box	Delete data should successfully remove	Delete data should successfully removed
05	Test Delete Page	White Box	Delete data should successfully remove	Delete data should successfully removed

Table 8.1 Summary of Testing 3