

**Student Information System
Integrated with Google Classroom
for Nalanda College**

**S.A.D.R.Lakshitha
2021**



Student Information System Integrated with Google Classroom for Nalanda College

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

**S.A.D.R.Lakshitha
University of Colombo School of Computing
2021**



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: S.A.D.R.Lakshitha

Registration Number: 2018/MIT/046

Index Number: 18550468



Signature: _____

Date: 2021-11-30

This is to certify that this thesis is based on the work of
Mr./~~Ms.~~ S.A.D.R.Lakshitha
under my supervision. The thesis has been prepared according to the format stipulated and is
of an acceptable standard.

Certified by:

Supervisor Name: Dr Thushani A Weerasinghe



Signature: _____

Date: 30-11-2021

ABSTRACT

During the Covid-19 pandemic situation in Sri Lanka around March 2020, most schools started teaching online to their students using online applications. Most of them choose to use freely available software packages. Nalanda College in Sri Lanka has also started its online studies using the Google classroom application.

However, not having a proper student management system is the main issue for all schools to handle the pandemic online sessions and their school administration. Nalanda College is also facing the same problem with its daily administrative work.

This system was started to support Nalanda College's administrative works to overcome most manual jobs. This system will help the college administration from students registration to their graduation in Grade 13. The system is designed to generate the admission card when a student is schooling. When he completes his education, the system will generate his leaving certificate according to all his school activities.

Also, as the college is using the Google classroom application for their online sessions, this system will help them create timetables without getting through the manual processes. As this system is integrated into the Google API, reports related to conducted sessions will also generate through the system.

HTML, CSS, Bootstrap, JavaScript, jQuery languages were used for front-end developments while SQL Server 2019 with ADO.Net was used in backend implementation and centralized database management of the project. MVC architecture is used to manage development easily while using visual studio 2019 combined with C# as the programming language. The incremental waterfall methodology is used to complete the project as it suits best over other methodologies in the completion of the project.

The main objectives of the project are increasing the productivity and efficiency of the school administration, keeping records in a centralized database, increasing the report generation accuracy were able to achieve in the project.

According to the suggestions received during the survey, several UI changes were carried out to get a better user experience. Also as per the feedback received the logic of the attendance report was reevaluated and carried out new development to fix the issue. The project will be extended for the next step of development according to the management decision of payment gateway integration and parental module.

ACKNOWLEDGMENTS

I would like to use this opportunity to express my gratitude to everyone who supported me in various ways to make this project a success.

First and foremost, I would like to give my deepest gratitude to the University of Colombo School of Computing for giving me this great opportunity to obtain a Master's degree. Also, I would like to thank the lecture panel and staff of the University of Colombo School of Computing for helping us and precisely guiding us.

Very special gratitude needs to be given to my project supervisor Dr Thushani Weerasighe for extensive assistance, without which the completion of this project would have been extremely complicated.

Also, my special thanks go to Mr Thilak Waththuhewa (The Principal), Mr Dinindu Alwis, Ms Gayathri Srinammuni, Mr Indika Nwarathna, and all the staff of the Nalanda College for the great support that they have given for this project.

Finally, I would like to thank my family members and friends of ISM Apac (Pvt) Ltd and everyone else who helped me succeeds with the project work while I engaged in office work.

TABLE OF CONTENTS

Abstract.....	iv
Acknowledgments.....	v
Table of Contents.....	vi
List of Figures.....	ix
List of Tables.....	x
List of Acronyms.....	xi
1 Chapter – Introduction.....	1
1.1 Project Overview.....	1
1.2 Motivation.....	2
1.3 Goal and Objectives.....	2
1.4 Project Scope.....	3
1.4.1 Out of the Scope.....	5
1.5 Feasibility Analysis.....	5
1.5.1 Technical Analysis.....	5
1.5.2 Financial Analysis.....	5
1.6 Structure of the thesis.....	6
1.7 Summary.....	6
2 Chapter – System Analysis.....	7
2.1 Current System Analysis.....	7
2.1.1 Project Background.....	7
2.1.2 Use Case Diagram of the current process.....	9
2.2 Fact-Finding Methodologies.....	10
2.2.1 User Interviews.....	10
2.2.2 Observations.....	11
2.2.3 Inspecting Existing Documents.....	13
2.3 Functional Requirements.....	15
2.4 Non-Functional Requirements.....	16
2.5 Review of Similar Systems.....	17
2.6 Summary.....	21
3 Chapter – System Design.....	22
3.1 UML Diagrams.....	22
3.1.1 Use Case Diagram.....	22
3.1.2 Use Case Narratives.....	23

3.1.3	ER Diagram of the proposed system	26
3.1.4	Class Diagram.....	28
3.1.5	Sequence Diagram of the proposed system	29
3.1.6	Database Design.....	32
3.2	User Interface Design.....	34
3.2.1	Login.....	34
3.2.2	Student Registration.....	34
3.2.3	Teacher Registration	35
3.2.4	Student Promotion	36
3.2.5	Student Extra Activities	37
3.3	Summary	38
4	Chapter - Implementation.....	39
4.1	Related Technologies	39
4.2	High-level Architecture of the System.....	40
4.3	Modules of the system.....	41
4.4	Major Code Segment.....	42
4.4.1	HTML page sample	43
4.4.2	Js page sample.....	45
4.4.3	Save Student Enrollment.....	46
4.4.4	Print PDF Report.....	47
4.5	Implementation Environment.....	48
4.5.1	Hardware Environment	48
4.5.2	Software and Technologies.....	48
4.6	Summary	49
5	Chapter – Testing and Evaluation	50
5.1	Related Testing Types.....	50
5.1.1	Unit Testing	50
5.1.2	Integration Testing.....	50
5.1.3	System Testing.....	50
5.1.4	User Acceptance Testing	50
5.2	Test Cases.....	51
5.3	Test Results	53
5.4	User Evaluation	62
5.4.1	Results of the Testing.....	64
5.5	Summary	66
6	Chapter – Conclusion	67

6.1	Lessons Learnt.....	67
6.2	Future Improvement.....	67
7	References	69
	Appendix A – System Documentation	70
	Appendix B – User Documentation.....	79
	Appendix C – MIS Reports.....	95
	Appendix D – Test Results	99

LIST OF FIGURES

Figure 1.1: Project Scope.....	4
Figure 2.1: DFD of the current process	8
Figure 2.2: Use Case Diagram of the current process	9
Figure 2.3: Created classes in Google classroom	12
Figure 2.4: Inviting Students to join with classes, each year on Google classroom.....	12
Figure 2.5: Student registry - Manual document	13
Figure 2.6: Timetable - Manual document	14
Figure 2.7: Student attendance on online sessions - Manual document	15
Figure 2.8: Fekara – Student Admission UI	18
Figure 2.9: Fekara - Timetable generation UI	18
Figure 2.10: OpenSIS - Student Registration UI	20
Figure 2.11: OpenSIS – Timetable UI	20
Figure 3.1: Use case diagram of the proposed system.....	22
Figure 3.2: ER Diagram.....	27
Figure 3.3: Class diagram	28
Figure 3.4: Sequence diagram of student registration	29
Figure 3.5: Sequence diagram of teacher registration	30
Figure 3.6: Sequence diagram of student class promotion	31
Figure 3.7: Sequence diagram of timetable generation	32
Figure 3.8: DB diagram of proposed system	33
Figure 3.9: User Interface of Login screen	34
Figure 3.10: User Interface of Student Registration	35
Figure 3.11: User Interface of Teacher Registration	36
Figure 3.12: User Interface of Student Promotion creation.....	37
Figure 3.13: User Interface of Club Membership Creation	37
Figure 4.1: MVC Architecture.....	39
Figure 4.2: High-level Architecture.....	40
Figure 4.3: RAD Methodology	40
Figure 4.4: Module interaction of the system	42
Figure 4.5: MVC file structure.....	43
Figure 4.6: HTML code of Teacher Information.....	44
Figure 4.7: Code segment of a Js file.....	45
Figure 4.8: Code segment - Save enrollment.....	46
Figure 4.9: Code segment - Print PDF report	47
Figure 5.1: Survey - Google Form page1	62
Figure 5.2: Survey - Google Form page2	63
Figure 5.3: Survey Responses Q1-3.....	64
Figure 5.4: Survey Responses Q4-8.....	65

LIST OF TABLES

Table 2.1: Interview 1	10
Table 2.2: Interview 2	11
Table 2.3: Interview 3	11
Table 3.1: Use case narrative of teacher registration	23
Table 3.2: Use case narrative of student registration	23
Table 3.3: Use case narrative of student's class promotion	24
Table 3.4: Use case narrative of Student's extra-curricular activity	24
Table 3.5: Use case narrative of class timetable generation	25
Table 3.6: Use case narrative of character certificate generation	25
Table 3.7: Use case narrative of online lecture report generation	26
Table 5.1: Test Cases	53
Table 5.2: Test Results.....	61

LIST OF ACRONYMS

API- Application Programming Interface

LMS- Learning Management System

HTML – Hypertext Markup Language

CSS – Cascading Style Sheet

SQL – Structured Query Language

RAD – Rapid Application Development

RAM – Random Access Memory

ER – Entity Relationship

ERP – Enterprise Resource Planning

UI – User Interface

ASP – Active Server Pages

IDE - Integrated Development Environment

IIS – Internet Information Services

MVC – Model View Controller

UAT – User Acceptance Testing

UML – Unified Modeling Language

SDLC – Software Development Life Cycle

DB – DataBase

GUI – Graphical User Interface

URL - Uniform Resource Locator

LINQ - Language-Integrated Query

1 CHAPTER – INTRODUCTION

Nalanda College, Colombo is a Buddhist school that operates under the ministry of education. Nalanda college provides many facilities such as science and computer laboratories, lecture halls, auditoriums, hostels, etc., for their students.

This chapter of the project describes the current situation of the student management process of Nalanda College. The chapter mainly focuses on its related issues and steps that need to take to overcome them.

1.1 Project Overview

As a result of the COVID19 outbreak, the management of Nalanda college decided to change their learning practice to online with an open-source platform named Google Classroom, after analyzing a few products such as Google Meet, Microsoft Teams & Zoom with the support of their Old Boy's Association. They have started using G-Suit as it offers free service for non-profitable companies. It comes with some attractive features such as free email addresses for all the students and teachers, unlimited G-Drive space, the Google Classroom package, and Google Meet to conduct online sessions for up to 250 students.

At the moment, the school's management is creating master data of classes, teachers, and students on the Google classroom application manually by referring to manual class registries and all other administrative paper documents.

Switching to Google Classroom (Etherington, 2017) and Google Meet (Javier Soltero, 2020) helped the school to streamline its day-to-day teaching, assignment submission, and grading processes during the outbreak. The students and staff were trained to use G-suit for their school activities.

But, due to the lack of a proper Student Information System, the management staff members including the school principal, facing many issues even after implementing the above online project. As they are currently using a manual process in their administrative tasks, including data feeding of student registrations, index no generations, class promotions, timetable creation, etc. Management has also found many more duplicated index numbers in the manual process.

As Google Classroom is not supporting customized reports, teachers are now facing issues with generating attendance sheets for their online sessions. Attendance is a required feature in the school process. Therefore, teachers create attendance details manually and report to the principal soon after having online sessions on Google Meet.

The principal is currently looking for a complete student information system that can be integrated with the e-learning platform (which is now used by their staff and students) to manage all the administrative work and reduce their repetitive tasks.

As the Google Classroom application does not allow for class promotions, the management must involve in the below process each year to add students for classes on the Google application to conduct online sessions.

1.2 Motivation

Due to the lack of a proper student information system, the school faces several issues when using Google Classroom for their administrative works on an online teaching facility.

Duplication of student index numbers is a significant issue faced in the current manual process of the school. Student data, teacher data, class teacher data, and subject-related teacher data are all managed manually by the school.

Since student class promotion is not supported in the google classroom application, during student's promotion to the next grade in each year, all classes need to be removed from classroom application and need to repeat entering all class information and student data each year. An annual repetitive manual process would remain to administrative staff if the school did not integrate the google classroom application with a proper student information system.

Since google classroom is not providing useful reports regarding online sessions, the principal of the school is now in trouble when collecting information regarding student attendance for each session and monitoring conducted sessions. The principal is also facing issues when presenting their attendance data in charts to the Ministry of Education. Mainly he is in huge trouble regarding the data accuracy since he has no way to collect online session data in one centralized place by using the Google Classroom application.

Also, the timetable creation has got conflicted due to teachers are working from home. They are sharing a lot of drafted timetables here and there to make a proper timetable for each grade. Due to a lot of sharing and conflicts, the time-tabling process is time-consuming for teachers and grade heads. They involve in a lot of paperwork as well.

1.3 Goal and Objectives

The project's goal is to provide a fully automated Student Management platform that is integrated with their e-learning platform.

The following objectives were defined to achieve the above goal.

- Reduce the manual work of the school administration, which consumes more unnecessary time in its process.
- Aid office assistants, teachers, and the principal of the school by reducing manual paper works in report generation, approval cycles, registry updates.
- Resolve the issue of having duplicated index numbers in student registration. Data will be stored centralized and updated in real-time to Google classroom as well.
- Reduce the time consummation and increase the data accuracy of manual attendance calculation by an automated process, which will be integrated with Google Meet and G-suit.

- Increase the efficiency of creating timetables and sharing them with students using google calendar.
- Increase the efficiency of students' class promotions each year without interacting with google classroom applications.
- Keep on recording students' past grade information without getting to lose any data.
- Increase the efficiency of report generation by real-time data synchronization of online sessions, conducted sessions, session cancellation, and upcoming online school sessions, etc., to the school principal.
- Increase the efficiency of managing student extra-curricular activities, student class promotions, students' leaving certificate creation, and report generation by streamlining and simplifying tasks related to them.

1.4 Project Scope

The system is mainly focused on aiding office staff, grade heads, teachers, and the school principal on their related administrative tasks with students and classes. The system will help the office staff to manage all student-related data from their registration with an automatic generation of index number facility. The student registration will be synchronized to the google classroom application via Google API.

Student's class promotions for each year will be handled through the system and it will be integrated into the google classroom via Google API to make new classes each year on the google classroom application. The system is designed to keep records of all student's promotional details, while google classroom only gets its updates.

The creation of grade timetables will be handled through the system according to class availability by grade head-teachers. The teacher availability should automatically indicate by the system. The created timetables will be updated in real-time to students' and teachers' Google calendars via an API synchronization. Session cancellation is also handled through the proposed system, and it will notify to assigned teacher and students as mentioned above. The information related to sessions conducted through the Google Meet application will be entered real-time into the system via an API synchronization.

A dashboard view of the online schedule, teacher schedule for past months, future months, and happening now will be available for the managerial role of the proposed system.

All reports related to google classroom sessions and students' attendance are generated through the system by the management.

The students' leaving certificate generation according to their extra-curricular activities will be available on the system for school office staff and it will need to get final approval from the principal of the school. The approval cycle is automated using the system.

Processes of the above-mentioned scope will go as follows. The administrative staff of the school and the managerial roles such as principal, vice principals, and grade head-teachers will be involved in processes in Figure 1.1.

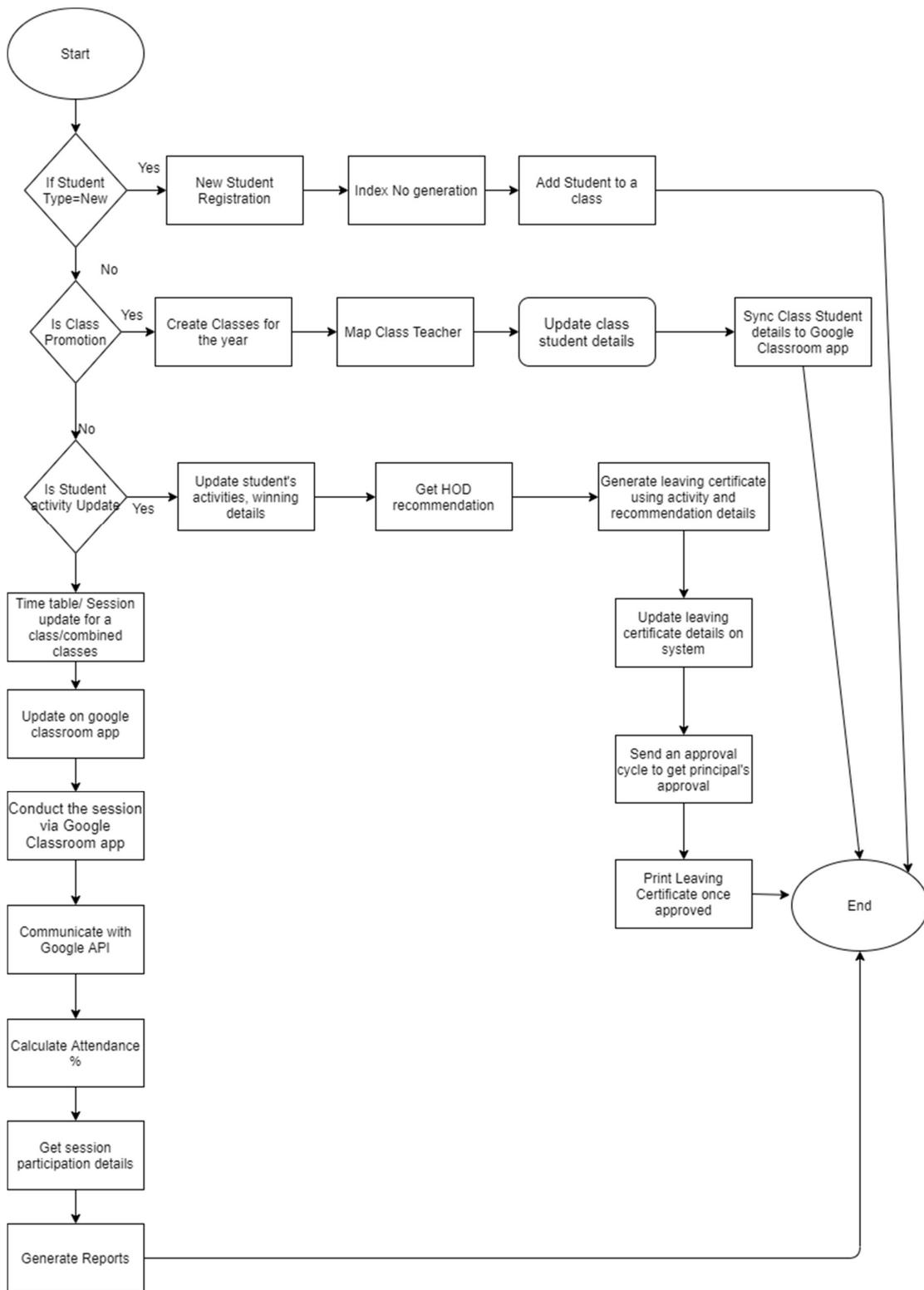


Figure 1.1: Project Scope

1.4.1 Out of the Scope

Since students are already comfortable using Google Meet and Google Classroom applications when joining online sessions and it's freely available for school management, the proposed system is not focusing on LMS-related functions such as assignment upload, video upload, student forums, and assignment grading.

The system is not visible or accessible to students at the school. They will only get access to the G-Suit where they can access Calendar, Classroom, and Meet applications for their e-Learning activities as its already executing perfectly by the school.

The proposed system is not aimed at eLearning features, as current students are familiar with the Google Classroom application, and it provides many user-friendly facilities to them, including unlimited storage.

The proposed system also does not focus on the classic offline timetable of the school as this is focusing to integrate with Google classroom. The application is only focusing on the online timetable and its functionalities.

1.5 Feasibility Analysis

1.5.1 Technical Analysis

As the product will be a web-based application, the following technologies will be used to implement the project.

- HTML
- CSS
- JavaScript
- SQL Server
- .Net technologies
- Diagram drawing tools
 - Visio
 - Microsoft Project
 - Draw.io

All the above technologies are freely available, it is clear this is technically feasible.

1.5.2 Financial Analysis

As the product is an online application, hosting its product will have a hosting cost. Also, the school office staff will need devices to access the application.

As the application is integrated into freeware software, it will not cost the product.

Besides the associated cost, there will be many benefits for the school for all their student management and online teaching. Especially the extra effort associated with google classroom will be significantly reduced and reports generating will be automated with the product.

According to the above, it is obvious that the project is financially feasible.

1.6 Structure of the thesis

- Chapter 1: Introduction
 - This will include a brief introduction to the project.
- Chapter 2: System Analysis
 - The chapter will include analysis, a review of similar systems, and a comparison of alternative design strategies.
- Chapter 3: System Design
 - The chapter will include proposing system design details including use cases, ER diagrams, Class diagrams, sequential diagrams, and proposed UI designs.
- Chapter 4: Implementation
 - The chapter will discuss important codes and the test plan of the developed system.
- Chapter 5: Evaluation
 - The chapter will be discussed whether the project objectives were satisfied and if not, the reasons for them. Lessons learned during the project and failures and reasons for failures.
- Chapter 6: Conclusion
 - The chapter will include the work indicating a summary of the results of the project.

1.7 Summary

The first chapter of the thesis discussed the basics of the entire project implementation. It presented the current problem, which is manual processes. Hence, the utmost aim of this project is to supply a feature-rich digital solution to manage limited resources efficiently and effectively in the Nalanda College.

2 CHAPTER – SYSTEM ANALYSIS

This chapter describes the analysis of the Student Management System integrated into Google classroom to streamline the administrative work related to the office staff, grade headteachers, class teachers, and the principal of Nalanda College, Sri Lanka. The project will help to overcome the issues in the lack of a proper Student Information System.

2.1 Current System Analysis

As the process of school is a manual process, they are also facing several issues when connecting with the Google Classroom application for e-learning.

As a solution for problems at school, the project needs to be is designed to manage the following areas.

- Staff details.
- Classroom details.
- Subject Details.
- Timetable details.
- All kinds of student details from the registration to their graduation.
- Manage students' leaving certificate generation using their activities.
- Report generation.
- Synchronizing data with Google classroom.

2.1.1 Project Background

Office assistants are the people who mostly interact with the master data of the school administration. All collected master data such as class details, subject details, teacher information, student information need to complete by them to begin the project.

The data flow diagram of the process is as follows.

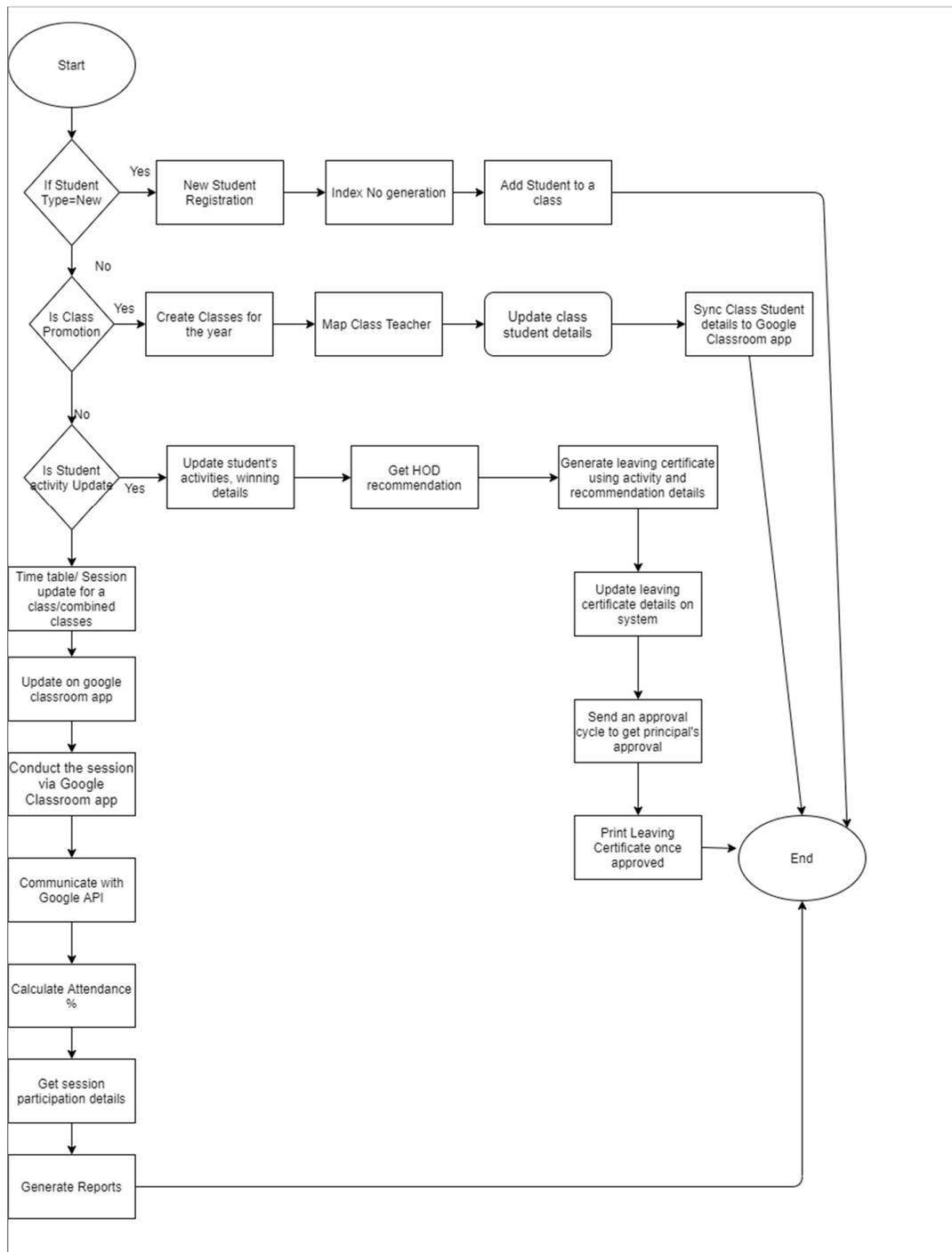


Figure 2.1: DFD of the current process

As per the above Figure 2.1, all new students are registered to the system by an office assistant. Once the student details are recorded in the system, it will automatically generate a new index number for the student. Grades and Classes were also recorded in the system by the office assistants. Teacher information also needs to record in the system. Some teachers are attached to grades as grade headteachers. Any other teacher or the same grade headteachers are attached to classes as class teachers. Other teachers who are not attached as grade heads or class teachers should be able to teach sessions. Also, class teachers or grade heads should be able to teach sessions to students.

Timetables are created by grade headteachers and should be able to inform class teachers about class sessions. Subjects are attached to classes as compulsory subjects and additional subjects. Some of the classes are now conducting sessions as “combined sessions”, by combining more than two classes at once. Students are selecting their additional subjects as per their interests. Timetables and student promotions need to synchronize with the Google Classroom application for the e-learning facility at the school.

This system needs to facilitate report generation features for grade headteachers and the principal for student participation in online sessions.

Students should promote to the next grade annually and their grade head teachers and class teachers also change according to the promotion. Student promotions need to be recorded properly in the system, as it is very much important for leaving certificate generation at the end of the student’s school life. To generate the leaving certificate of the student, the system should properly record the student’s extra-curricular activities and annual overview of the student by their class teacher. Extra-curricular activities are handled by grade headteachers and office assistants.

2.1.2 Use Case Diagram of the current process

This project is designed to manage student information and timetables while connecting with the Google Classroom application.

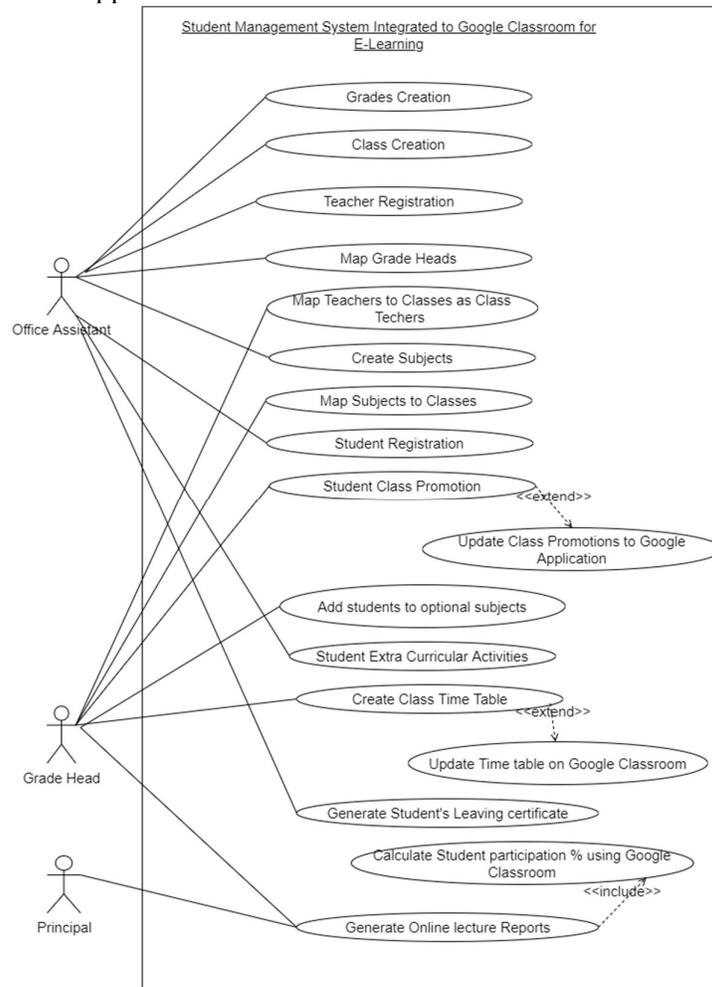


Figure 2.2: Use Case Diagram of the current process

As per the ave Use Case Diagram in Figure 2.2, the Office Assistants of the school are managed teacher information and student registrations. Grade heads are appointed by the principal, and it also updates to manual books by office assistants. Grade headteachers are allocated annually by the principal and it also updated to books by office assistants. Student class promotions are handled by grade headteachers, and the promoted classes of each year need to be recorded on the system for a student leaving certificate generation. All student records of their extra-curricular activities, prize, and colour winning details, scholarship details are managed through the project and be used in the student leaving certificate generation.

Class timetables are designed and updated by grade headteachers, and the principal is involved in report generation of conducted and cancelled online sessions. The principal is also considering student participation in online sessions.

2.2 Fact-Finding Methodologies

To gather requirements from users for implementation of the project used as follows.

- User Interviews
- Observations of the process
- Inspect existing documents in the school

2.2.1 User Interviews

The principal, office assistants, grade headteachers, class teachers are the users of the project as per the process of the school. One staff member from the office assistant category, two grade headteachers, and the principal were interviewed to find facts for the related process of the existing system during the requirement gathering process.

The summary of the interviews is shown below.

Interview 1:

Interviewer	S.A.D.R. Lakshitha
Interviewee	Principal of the school
Venue	Nalanda College, Colombo
Date	15-18 th September 2020
Time Duration	1 Hour
Information & Decisions	<ul style="list-style-type: none"> • The problems that occur in the existing process. • Solutions to overcome the issues of the current manual process. • Reports need to be generated to automate the principal's tasks.

Table 2.1: Interview 1

Interview 2:

Interviewer	S.A.D.R. Lakshitha
Interviewee	Office Assistant
Venue	Nalanda College, Colombo
Date	05-10 th October 2020
Time Duration	3 Hours
Information & Decisions	<ul style="list-style-type: none">• Existing manual process in student management.• The problems that occur in the existing process.• Solutions to overcome the issues of the current manual process.

Table 2.2: Interview 2

Interview 3:

Interviewer	S.A.D.R. Lakshitha
Interviewee	Grade 11 Head Teacher Grade 10 Head Teacher Grade 8 Head Teacher
Venue	Nalanda College, Colombo
Date	05-10 th October 2020
Time Duration	3 Hours
Information & Decisions	<ul style="list-style-type: none">• Existing manual process in student management.• The problems that occur with the existing process.• Solutions to overcome the issues of the current manual process.

Table 2.3: Interview 3

2.2.2 Observations

The observation was a very supportive method to understand the current business process. It gave a good understanding of their office tasks, workflows, and their communication methods related to the process.

2.2.3 Inspecting Existing Documents

Following documents were found from the school as their current documents.

- Student Registry.

Figure 2.11 refers to a page of a manual school class registry. Student registration nos are recorded here and the school started creating a google classroom account by using these kinds of registry pages.

අංකය අංකය අංකය අංකය	අංකය අංකය අංකය අංකය	නම නම නම නම	පිහිටීම පිහිටීම පිහිටීම පිහිටීම			අංකය අංකය අංකය අංකය
			අංකය අංකය අංකය අංකය	අංකය අංකය අංකය අංකය	අංකය අංකය අංකය අංකය	
24098		Tehan P. kalupahana	06	05	08	1
24115		Pavith T. Rodrigo	06	12	03	2
24147		Bevin J. kasthuriarach	06	07	11	3
24152		R. Fibbisheka S. Dammun	06	09	24	4
24156		M.M. Senula Lanatolle	06	04	12	5
24159		Sathindra J.S. Siriwarda	06	03	01	6
24160		Sandila N. Piyasena	06	09	04	7
24167	x	Thisal B. Thandaviarach	06	10	30	8
24186		K.Vidas D. Perera	06	10	11	9
24189		K.D. Sandaru Nanayak	06	02	12	10
24190		Raveen M. Premaratna	06	09	26	11
24196	x	Sukitha Dharmasingha	06	09	07	12
24197		Lesandu Thilakaratna	06	02	02	13
24201		Sithil N. Silva	06	08	04	14
24202		D.Gauth Waidyaratna	07	01	30	15
24207		Kaviru G. Indralal	06	10	10	16
24219		Nevinda Gunasekara	07	01	15	17
24220		Ashin C. Rabel	06	12	31	18
24226		P. Akash Gunasekara	06	05	23	19
24227		Randeer S. Senarathna	06	03	12	20
24228		Savindu N. Epasingha	06	03	02	21
24241		H.K. Nuwanidu Ariyaratna	06	02	21	22
24269		W. Tharul K. Bothraju	06	11	23	23
24406		Thubiru S. Yapa	06	08	31	24
24410		Nikittha Bandara	06	06	01	25
24427		Nethula N. Weerasinghe	06	05	18	26
24430		E.W. Chenitha N. Ekanayake	06	11	06	27
24573		J. Sanjula Alahakoon	06	07	12	28
24579		E. Mervan Vijewardana	07	01	06	29
24581		B. Thenula Thejana	06	04	04	30
24582		Giadeesha Sandeep	06	10	03	31
24584		U.G. Risinu Randitha	06	10	11	32
26758		W. Chenitha De Zoysa	07	01	04	33
26760		Nelindu M. Dhampahala	06	07	07	34
26780		H. Rithithara Fernando	06	06	10	35
26789		G. Sahan Indrachapa	06	03	23	36
26796		J.K. Dayan Helitha	06	02	12	37
26811		M.A. Sathis Samarathna	06	03	27	38
26833		Namith D. Muthumala	06	03	23	39
26863		G.H. Kavindu Kalpath	07	01	11	40
27587		E. Sandaneth L. de Zeis	06	04	25	41
27863		S.N.W. Vidana	06	10	15	42
		Sethi				43

Figure 2.5: Student registry - Manual document

- Class timetable.

Figure 2.12 refers to an online timetable generated manually by a grade headteacher.

<p style="text-align: center;">2021 - 9 ශ්‍රේණිය Online ඉගෙනුම් වැඩසටහන 2021.02.15 - 2021.02.19 දක්වා කාල සටහන</p>					
දිනය	වේලාව	විෂය	පන්ති / පන්තිය	ගුරුභවතාගේ නම	
2021.02.15	පෙ.ව.6.00 – 7.00	බුද්ධාගම	C, D / F, H	සූජ්‍ය හේවැස්සේ අමීත හිමි	
			B, G	සූජ්‍ය අමරවල සෝමරත්න හිමි	
			A, E, I	සූජ්‍ය බැද්දේවෙල ධම්මකිත්ති හිමි	
	පෙ.ව.7.30 – 8.30	විභූ	A, B, G, H	දිනුෂ උපසේන මයා	
			C, D, E, F	සජීවා ආරච්චි මයා	
		තැවුම්	A, B, C, D	අමීලා ජීවනී මයා	
		පෙරදිග සංගීතය	A, B, G, H	කල්ප උදයප්‍රිය මයා	
			C, D, E, F, I	K.L.D.C. දමයන්ති මයා	
		අපරදිග සංගීතය	A, E, F, G, H, I	සජීවනී රත්නාසක මයා	
				දිනිකී ප්‍රනාන්දු මෙහෙවියා	
	තාට්‍ය හා රංග කලාව	A, B, G, H	ඩීලානි නිලංගා මයා		
		C, D, E, F	අනෝමා කුමාරසිංහ මයා		
2021.02.16	පෙ.ව.6.00 – 7.00	දෙමළ	B, C / D, E	ප්‍රහා නිරෝශිනී මයා	
			ඉංග්‍රීසි	H	දිලිනි සමරසිංහ මයා
				I	දර්ශිනී අලක් මයා
				A, F	H.P.P. දිල්ලික්කි මයා
	පෙ.ව.7.30 – 8.30	දෙමළ	A / F, G / H, I	ප්‍රහා නිරෝශිනී මයා	
			ඉංග්‍රීසි	B, C	දිසිකා සමරවීර මයා
				D, E	මුදිතා කල්දේරා මයා
					මුදිතා කල්දේරා මයා
2021.02.16	ප.ව.6.30 - 7.30	ඉංග්‍රීසි	G	මුදිතා කල්දේරා මයා	
2021.02.17	පෙ.ව.6.00 – 7.00	ගණිතය	A	හිමාලි නානාසක්කාර මයා	
			B, I	ප්‍රදීප ජයරත්න මයා	
			C, E	වසන්ත වඩුගේ මයා	
			D, H	S.D. සමරනාසක මයා	
			G, F	නිරූපමා රත්නාසක මයා	

Figure 2.6: Timetable - Manual document

- Teacher and student attendance sheet.

Figure 2.13 refers to a student attendance summary report of grade 10 during a particular date range. This report is a manually created report by the grade headteacher. Data for the report is collected from each teacher, once a session is completed.

මාර්ගගත ඉගැන්වීම් සති සාරාංශය - 10 ශ්‍රේණිය						
2021.01.26 සිට 2021.01.29 දක්වා						
දිනය	වේලාව	විෂය	පංතිය/පංති	ඉගැන්වූ පාඨම	සිසුන් ගණන	ගුරුවන්ගේ නම
2021.01.26 (අභ්‍යන්තරව)	පෙ.ව. 6.00 - 7.00	ගණිතය	10 - A(E.M)	වර්ගමූලය	31/42	අනුරාධ පෙරේරා මියා
			10 - C/ F	වර්ගඵලය	59/83	නිමාලී නානායක්කාර මියා
			10 - D	පරිමිතිය	30/41	නිරුපමා රත්නායක මියා
			10 - E /G	පරිමිතිය	56/83	ඩී.එස්. සමරනායක මියා
			10 - H	පරිමිතිය අභ්‍යාසය	22/42	ප්‍රදීපා ජයරත්න මියා
	පෙ.ව. 7.30 - 8.30	සිංහල	10 - A/B/I	සතර කන් මන්ත්‍රණය	55/127	එස්.ඒ.ප්‍රියංගනී මියා (2021.01.30)
			10 - C/E	අක්ෂර ප්‍රභේද	27/82	ජී.බී.දයානන්ද මියා
			10 - D	සිංහල වර්ණ මාලාව - හැඳින්වීම	20/41	ඩබ්.ඒ. එරෙදි මියා (2021.01.30)
			10 - F		/42	වමින්ද ගුණරත්න මියා
			10 - G		/42	නිලුක්ෂි පෙරේරා මියා
		10 - H	අතීතයෙන් කතාවක්	36/42	පී.කේ.ආරච්චි මියා	
2021.01.27 (බදාදා)	පෙ.ව. 6.00 - 7.00	බුද්ධ ධර්මය	10 - A/C/ D	නම්දිමු අගයමු මහ සභ රුවනා	103/124	පුජ්‍ය හෙවෙස්සේ අමිත හිමි
			10 - B/E/H	නම්දිමු අගයමු මහ සභ රුවනා	93/126	පුජ්‍ය අමිත්වල සෝමරත්න හිමි
			10 - F/G/I	අභියෝග ජය ගත් සිද්ධත් ඛේමසතාණේ	61/121	පුජ්‍ය බැද්දේවෙල ධම්මකිත්ති හිමි
	පෙ.ව. 7.30 - 8.30 (නාණ්ඩ විෂය 02)	විභූ	10 - A/B/C/F/G	Solias Mendis (Part 1)	35/46	දිනුෂ උපසේන මියා
			10 - D/E			ආනන්ද කාරියපිම් මියා
		නැටුම්	10 - B/C/F/G	රබන් කවි වල පසුබිම හා රබන් කවි	21/23	අමිලා ජීවනී මියා
			10 - A/B/C/F	අභ්‍යන්තර රාගය	26/36	කල්ප උදයප්‍රිය මියා
		පෙරදිග සංගීතය	10 - D/E/H/I	අභ්‍යන්තර රාගය	35/48	කේ.එල්.ඩී.සී.දයානන්ති මියා
			10 - A/B/ G/H	C clefs (Alto and Tenor) - 1	35/42	සජීවනී රත්නායක මියා
		නාට්‍ය	10 - A/B/C/ E	දෘෂ්‍ය කලා හා ප්‍රසංගීක කලා	28/41	ඩිලානී නිලංගා මියා
			10 - D/H/ I	සතර අභිනය හා වලන වර්ග	27/39	අනෝමා කුමාරසිංහ මියා
		සිංහල සාහිත්‍ය	10 - B/C/G/H/ I	සුඛාභිනය කවි විස්තර කිරීම	39/53	ඩබ්.ඒ.එරෙදි මියා
		ඉංග්‍රීසි සාහිත්‍ය	10 - A /B/H	To the evening star	23/36	දර්ශනී අලස් මියා

Figure 2.7: Student attendance on online sessions - Manual document

2.3 Functional Requirements

Following functional requirements need to be satisfied in the developed system to accept the product from the user's end.

Requirements of the Office Assistant:

- The system should facilitate changing the no of students in a class when required.
- Teacher profiles need to be managed through the system including their qualifications and teaching preferences.
- The system should be able to manage Grade Headteachers and Class teachers need to be separated each year.

- The system should facilitate generating the next index no automatically when a new student admission is completed.
- At the student admission completion or after the admission process, the student should be able to add to a class.
- The system should facilitate having an index no for every student and student's guardian details, sibling details.
- The system should be able to facilitate to capture the student image in the admission and when to search the student, the captured image needs to be visible to the user's end.
- The system should facilitate getting an indicator of whether the student character certificate is issued or not.

Requirements of Grade Headteachers:

- The system should facilitate the student's class promotion each year.
- The system should be able to facilitate managing students' final marks of each grade.
- Selected optional (Basket) subjects of students need to be managed through the system.
- Changing of student's selected basket subjects needs to facilitate to grade headteachers.
- Appointing class teachers and changing them need to be facilitated through the system.
- The system should be able to manage students' class promotions each year.
- Students' extracurricular activities should be managed and at the character certificate generation, it should automatically be filled according to data in the system.
- The system should facilitate generating the student's character certificate.
- The system should manage the conflicts of class timetable generation.
- The system should be able to generate timetables separately for physical classes and online classes.
- The online class and physical class allocation need to be separately managed through the system.
- The system needs to manage online timetable allocation.
- Once students are allocated to an online session, the system should facilitate updating the Google calendar of each allocated student and the teacher.

Requirements of the Principal:

- The system should facilitate getting a report of teachers' off time.
- The system should facilitate assigning teachers to different sessions in different classes (other than the timetable) when they are not assigned to any other class at a given time slot.
- The system should facilitate generating reports such as online session participant reports, online class allocation, teacher allocation.

2.4 Non-Functional Requirements

Following non-functional requirements need to be fulfilled throughout the system for a better user experience. (AltexSoft, 2019)

- Usability:

- The system needs to be well designed by fact accessing, fast loading, errorless with simple understandable interfaces to achieve the usability of the system. This may directly affect user attraction in the system as well.

Eg:

- Each request should be processed within 10 seconds.
- The site should load in 3 seconds.

- Security:

- The system records need to be secured as per the accessibility criteria of each functionality, database backup procedures.

- Portability:

- As the system needs to be a web-based application, the system needs to be portable to access and responsive for any kind of device such as PC, laptop, tablet, or mobile.

2.5 Review of Similar Systems

Following are some of the School Management Systems comparisons integrated to online learning platforms available on the internet.

<u>Product 1:</u>	
Product Name (; fekara.com, 2020)	Fekara
Product Logo	
Available Features	<ul style="list-style-type: none"> ● Admission & Enrolments ● Self-Enrolment ● Attendance ● Timetable ● Sessions and Batches ● Student Reporting ● Customize Report Templates ● Schedule Report generation ● Email/ SMS Reports ● Financial Reports & Summaries ● Management Reports ● Online Classes integrated with Zoom ● Meeting Creation within the app ● Meeting Link for Teachers ● Meeting Link for Students

User Interfaces

Figure 2.8 refers to the student admission UI of the Fekara product.

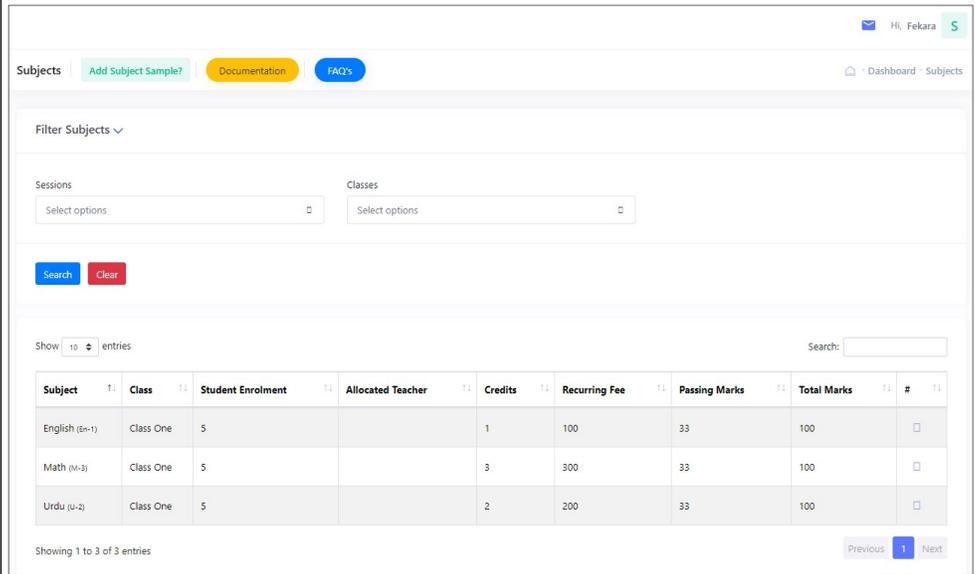


Figure 2.8: Fekara – Student Admission UI

Figure 2.9 refers to the timetable generation UI of the Fekara product. It gives edit, search, clear options for added sessions. Apart from them, it allows the user to send a message to guardians regarding the added session as a notification.

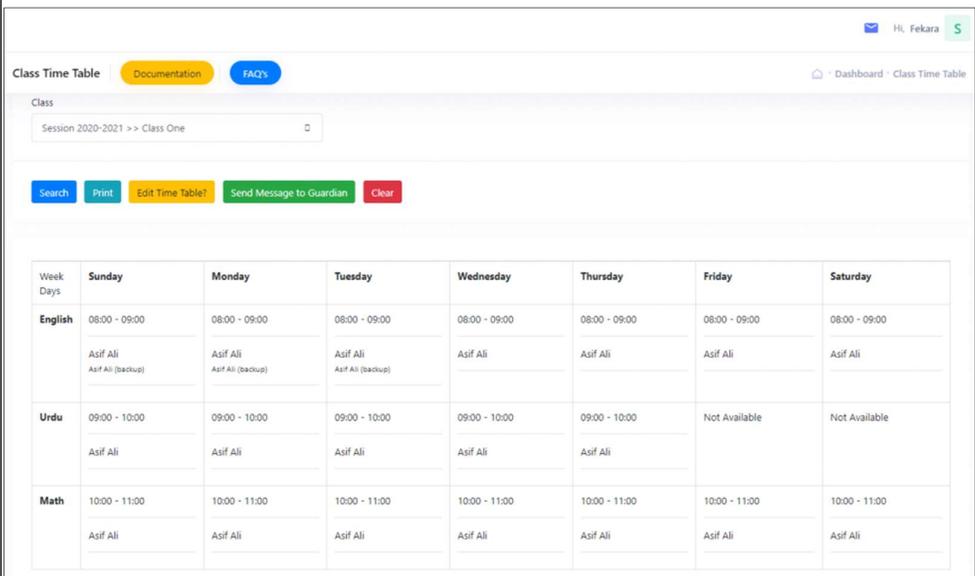


Figure 2.9: Fekara - Timetable generation UI

Price

Opt 1: Free for 50 students, 5 teachers with basic modules
 Opt 2: 600 students, unlimited teachers, basic and advanced features

<u>Product 2 :</u>	
Product Name (Anon., n.d.)	OpenSIS
Product Logo	
Available Features	<ul style="list-style-type: none"> • Student and parent portal • Teacher portal • Self-hosted • Student and Staff Demographics • Multiple language support • Reporting • Email messaging • Scheduling • Transcript's manager • Virtual classroom • Report card • Transcripts • Parent's portal • Grade-book management • Health records • School Calendars • Built-in messaging system

User Interfaces

Figure 2.10 refers to the student registration UI of the OpenSIS product. This UI allows the user to add a picture of the student at the registration process and when a student search, the picture also displays to the user.

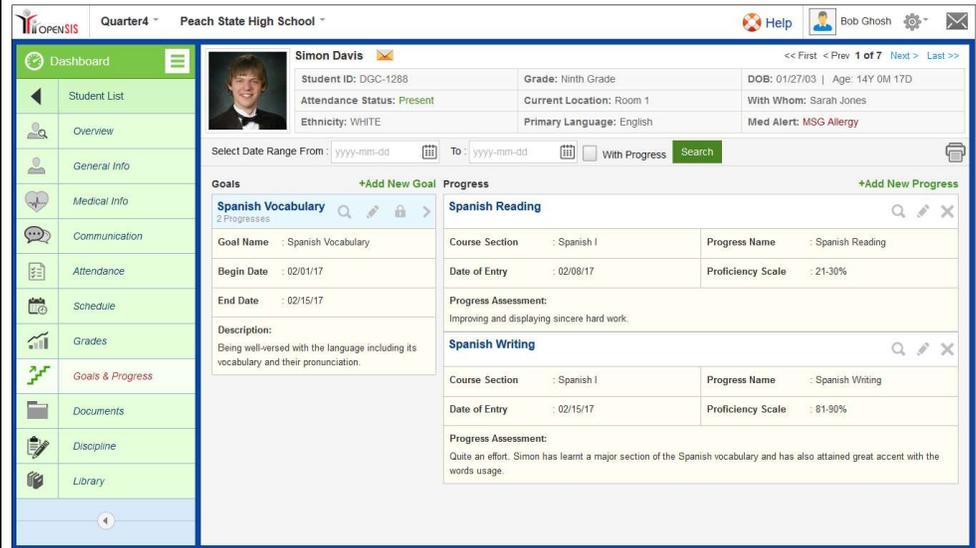


Figure 2.10: OpenSIS - Student Registration UI

Figure 2.11 refers to the timetable UI of the OpenSIS product. This UI provides the student count and its status to the user of each session.

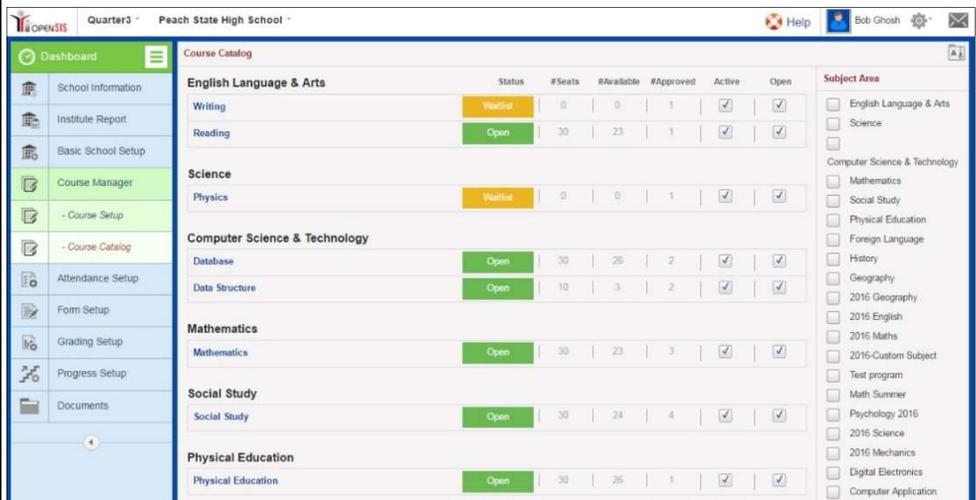


Figure 2.11: OpenSIS – Timetable UI

2.6 Summary

This chapter focused on providing the system analysis of the current process and similar applications related to the process in the market. The chapter provided a detail of the market product comparison as well.

3 CHAPTER – SYSTEM DESIGN

The proposed system is designed based on the user requirement and a detailed analysis of the gathered requirement. This chapter describes the architectural design of the analyzed and proposed Student Information System to Nalanda College, which will integrate their e-learning to Google Classroom application.

System design is the most crucial phase of any system development. The logical system design arrived at as a result of system analysis and is converted into physical system design. In the design phase, the SDLC process continues to move from the questions of the analysis phase to the how.

This chapter may focus on the Object-Oriented Process of the proposed system with its UML diagrams.

3.1 UML Diagrams

3.1.1 Use Case Diagram

A use case diagram is a behavioural diagram in UML diagrams, which is used to get understand the functional behaviour of the system. The use case diagram for the proposed system illustrates in below Figure 3.1.

Office Assistant, Grade Head Teacher, and the Principal act as main actors of the system.

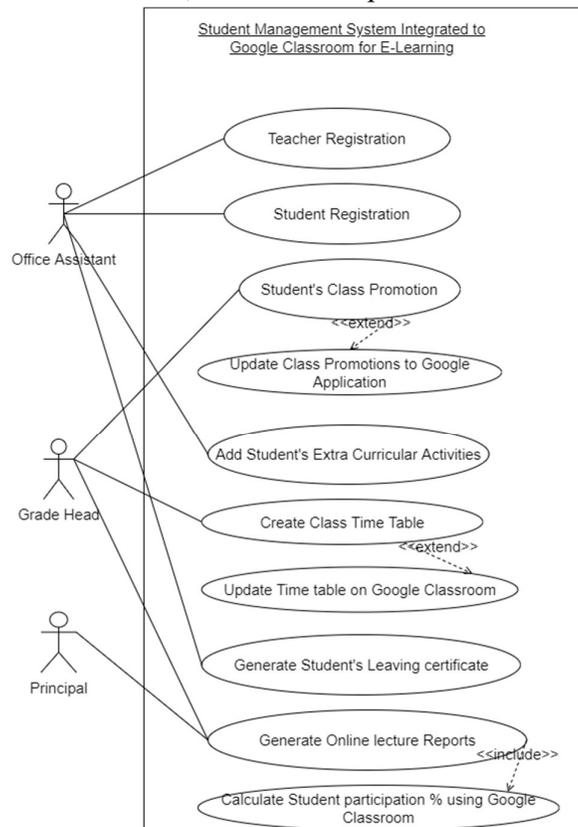


Figure 3.1: Use case diagram of the proposed system

3.1.2 Use Case Narratives

Use case 1: Teacher Registration

Use case	Teacher Registration
Actors	Office Assistant
Overview	
Master data related to teachers will be stored on DB	
Preconditions	
The user must log in to the system. Data should be entered correctly as per validation rules.	
The typical course of events	
<ol style="list-style-type: none"> 1. Fill the form with master data. 2. Save on the DB. 	
Postconditions	
Saved teachers will be visible on Class teacher mapping and timetable creation.	

Table 3.1: Use case narrative of teacher registration

Use case 2: Student Registration

Use case	Student Registration
Actors	Office Assistant
Overview	
Register new students at school.	
Preconditions	
The user must log in to the system. Data should be correctly entered into the system.	
The typical course of events	
<ol style="list-style-type: none"> 1. Fill in student details. 2. Save data on DB. 3. Generate the index number according to the sequence of its generation. 	
Postconditions	
Student data should be available for other function use. Index numbers should not be duplicated.	

Table 3.2: Use case narrative of student registration

Use case 3: Student's Class Promotion

Use case	Student Class Promotion
Actors	Grade Head Teacher
Overview	
Annual student's class promotions.	
Preconditions	

The user must log in to the system. The new academic period should be available in the system. Student records should be available.
The typical course of events
1. Select the academic period. 2. Select the grade and class. 3. Select a teacher as the class teacher. 4. Select students from last academic year, lower grade.
Postconditions
Students' grades and classes should be updated, and class teachers should be updated.

Table 3.3: Use case narrative of student's class promotion

Use case 4: Add student Extra-curricular activity

Use case	Add student Extra-curricular activity
Actors	Office Assistant
Overview	Save extra-curricular activities and award-winning details to the system for the use of generating the leaving certificate.
Preconditions	The user must log in to the system. Data should be correctly entered into the system. Students should be registered on the system.
The typical course of events	1. Select Student. 2. Add details of the activity. 3. Add the supervised teacher as the approver. 4. Add winning details. 5. Save data to DB.
Postconditions	Saved students should be available to allow online sessions.

Table 3.4: Use case narrative of Student's extra-curricular activity

Use case 5: Create a class timetable

Use case	Create class timetable
Actors	Grade-Head teacher
Overview	Creation of class timetables and synchronizing online schedules to Google classroom
Preconditions	The user must log in to the system. Classes, subjects, subject teachers should be correctly entered and available on the system.

The typical course of events	
1.	Select the class, date, time.
2.	Save data.
3.	Sync data with Google classroom for online sessions.
Postconditions	
Students should be able to join for online lectures and all session report should be able to generate.	

Table 3.5: Use case narrative of class timetable generation

Use case 6: Generate student's character certificate

Use case	Generate student's character certificate
Actors	Office Assistant
Overview	
Generate student character certificates based on extra-curricular activities they participated in.	
Preconditions	
The user must log in to the system. Students' extra-curricular details must be entered and approved.	
The typical course of events	
1.	Search the student.
2.	Get participation details.
3.	Generate document.
4.	Print the document.
5.	Block to generate the report again as it is a one-time process for a student.
Postconditions	
The office assistant should not be able to generate the report again.	

Table 3.6: Use case narrative of character certificate generation

Use case 7: Generate Online Lecture Reports

Use case	Generate online lecture reports
Actors	Grade-Head teacher, Principal
Overview	
	Generate a grade-wise report with participation count for online sessions.
Preconditions	
	The user must log in to the system. Google API should be integrated. Lecture schedule details should be available.
The typical course of events	
	1. Select the period of report generation. 2. Generate report. 3. Participation data receives through API and show on the report.
Postconditions	
Management decision making	

Table 3.7: Use case narrative of online lecture report generation

3.1.3 ER Diagram of the proposed system

The following Figure 3.2 refers to the ER Diagram of the proposed system.

Referring to the below-attached ER diagram, Students are attached to classes and their promoted details will be captured by the system. The school has grades and classes. Classes are attached to grades and subjects are also attached to grades.

Schoolteacher information is also managing and teachers are assigned to classes as class teachers and teachers are also assigned to several subjects to teach.

Lecture schedules are managed to report generation purposes. It should indicate online lecture sessions on the lecture schedule.

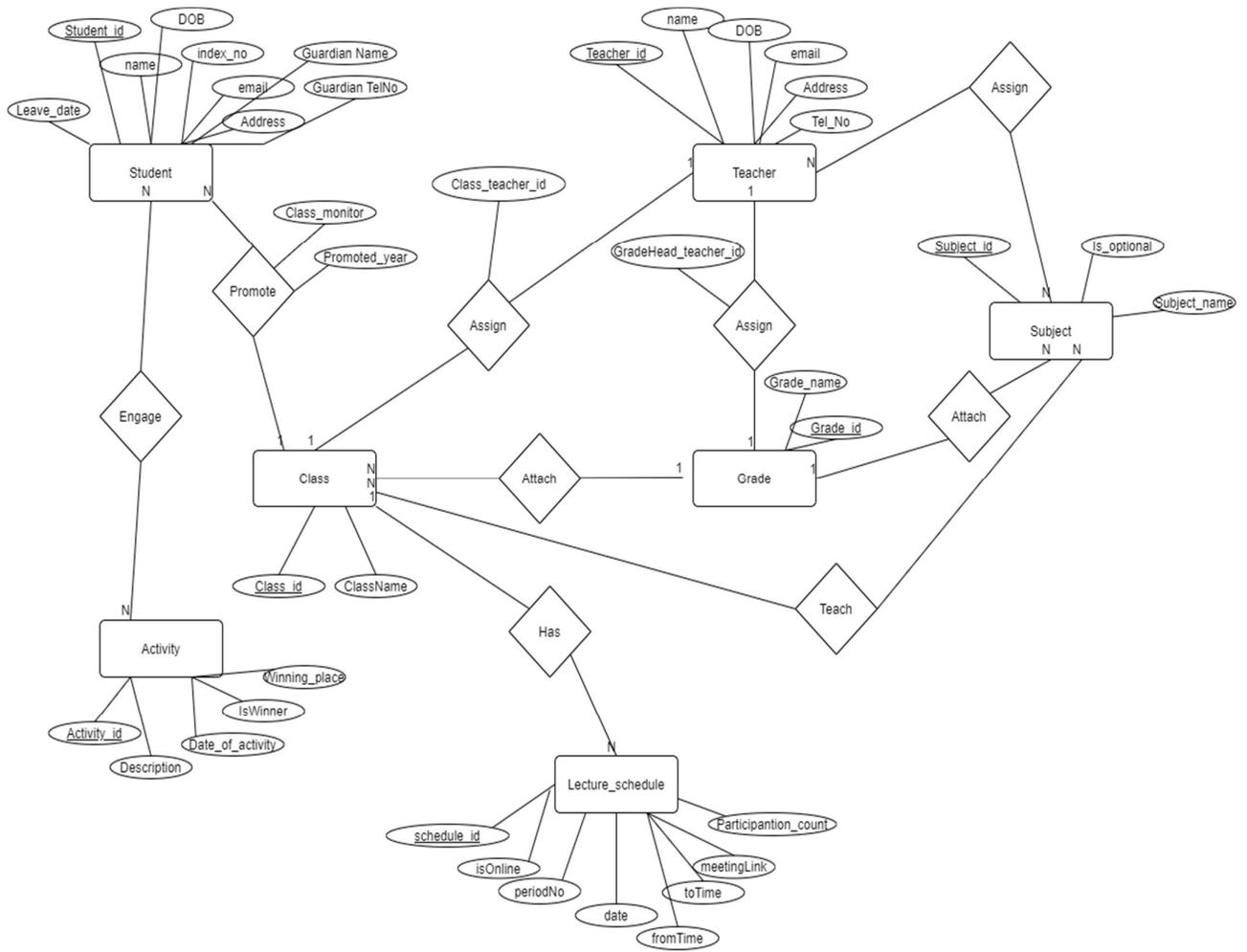


Figure 3.2: ER Diagram

3.1.4 Class Diagram

Figure 3.3 refers to the class diagram of the proposed system as follows. It describes the structure of the proposed system by indicating classes, attributes, functionalities, and their relationships.

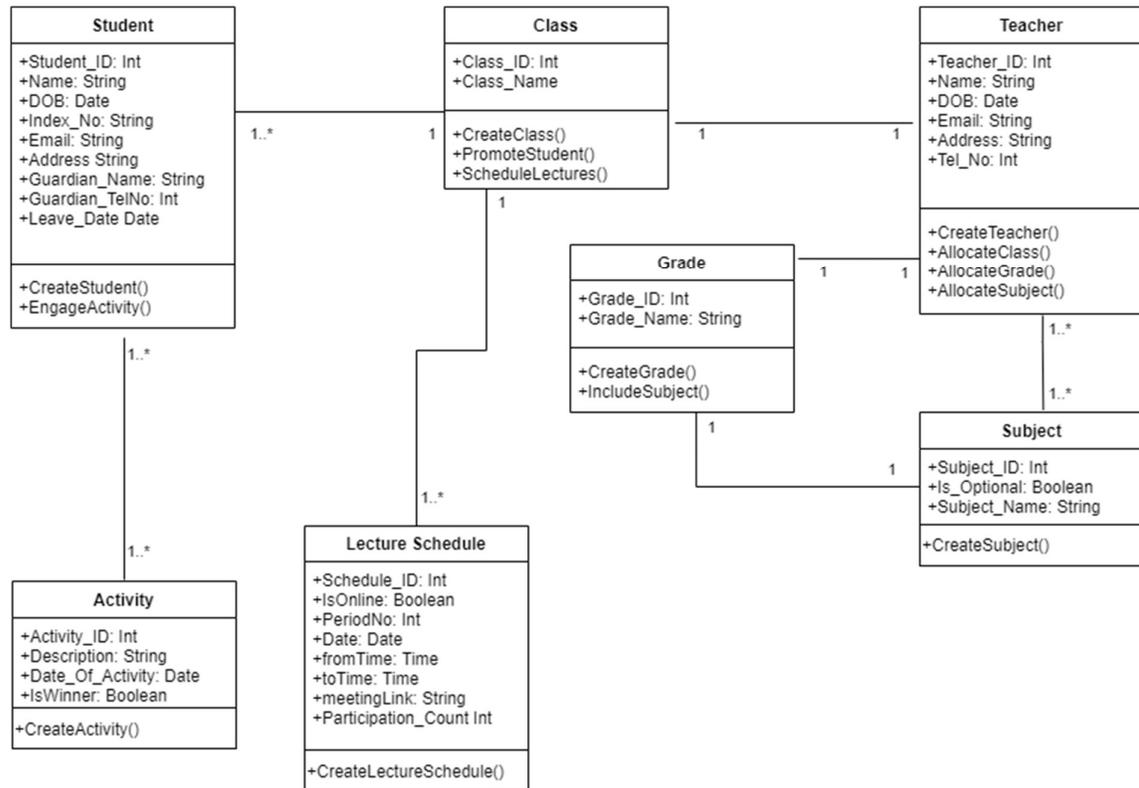


Figure 3.3: Class diagram

3.1.5 Sequence Diagram of the proposed system

Figures 3.4, 3.5, 3.6, 3.7, 3.8, and 3.9 refer to the sequence diagrams of the proposed system.

3.1.5.1 Sequence Diagram 1: Student Registration

Figure 3.4 refers to the sequence diagram of the student registration of the proposed system.

Once the office assistant entered data into the provided UI, a button action will send entered data to the controller to save it on DB. At the point of saving data, the controller will generate its index no. once after data is saved on DB, the assistant will receive the success message through the UI.

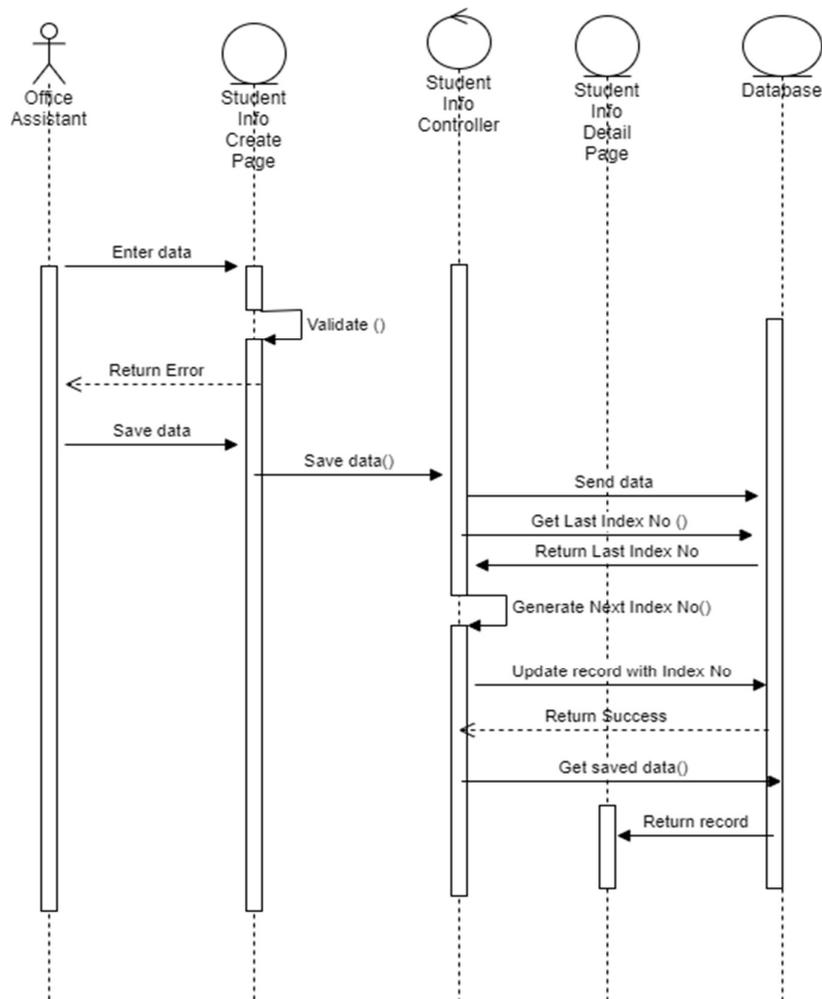


Figure 3.4: Sequence diagram of student registration

3.1.5.2 Sequence Diagram 2: Teacher Registration

The following Figure 3.5 refers to the sequence diagram of the teacher registration process of the proposed system.

When the office assistant entered data to the UI provided on the system correctly, data will pass to the DB through the page controller. Once data is saved on DB, the office assistant will get appear the success message on the UI.

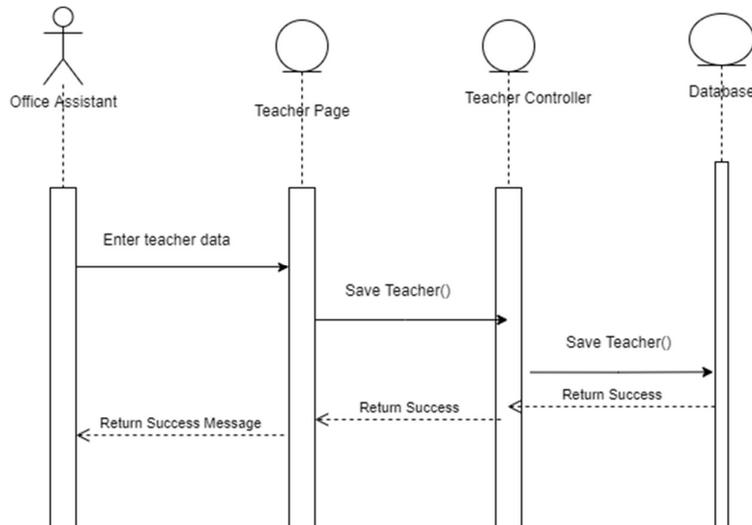


Figure 3.5: Sequence diagram of teacher registration

3.1.5.3 Sequence Diagram 3: Student Promotion and assign class teacher

Figure 3.6 refers to the annual student promotion and class teacher assigning process of the proposed system. The grade headteacher will be the user of the process.

The grade headteacher will select the new academic year and the promoting class using given controls of the UI. According to the selected class, the Grade headteacher will be able to select students from their last year grades. At the same time, UI will enable a selection of the class teacher to the selected class.

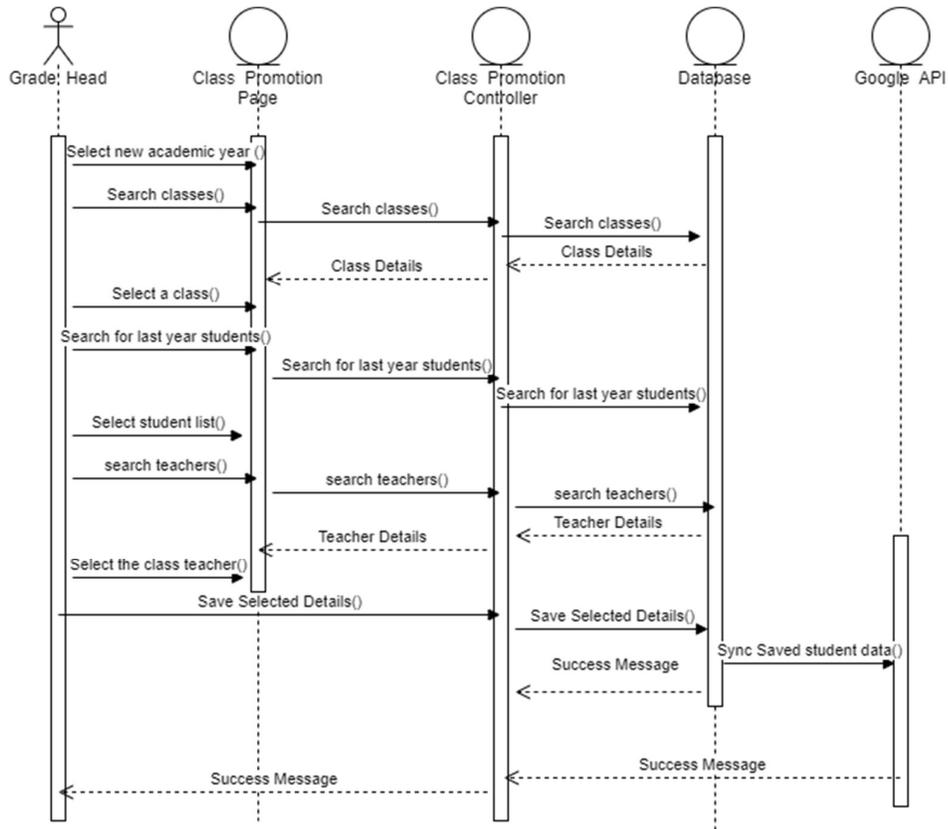


Figure 3.6: Sequence diagram of student class promotion

3.1.5.4 Sequence Diagram 4: Time-table Generation

Figure 3.7 refers to the sequence diagram of the timetable generation process of the proposed system.

The user of the process will be the Grade headteachers at the school. They can create lecture sessions for the selected class, for the selected time, and the selected date as per its availability.

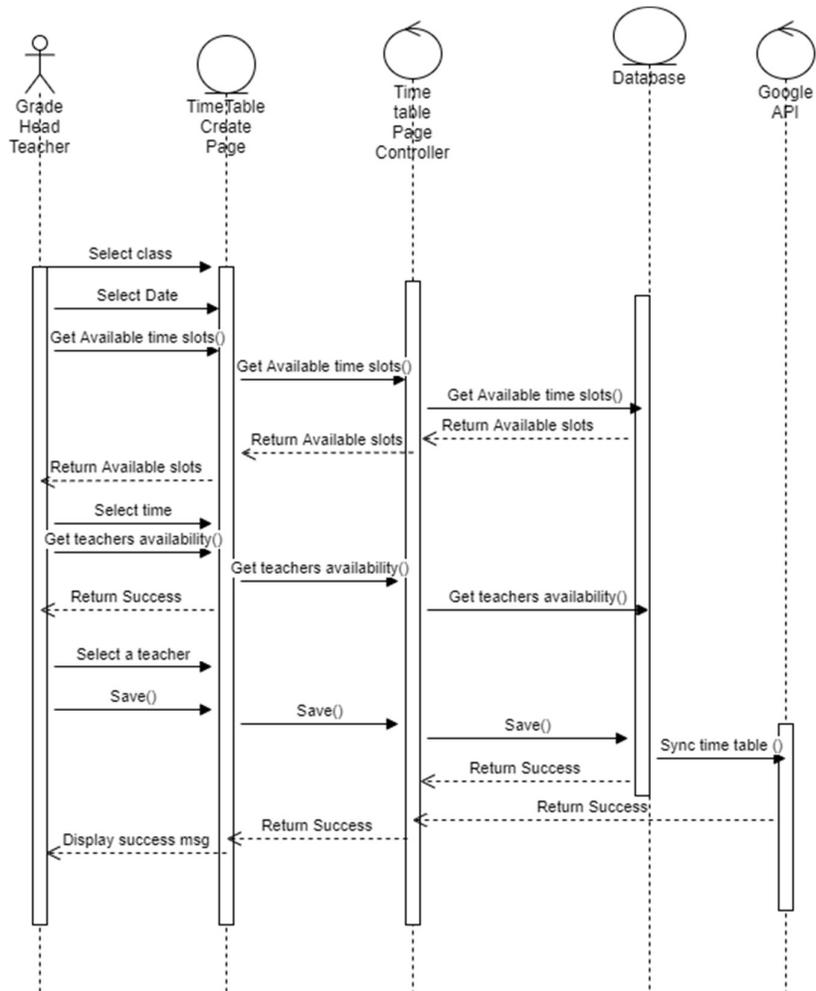


Figure 3.7: Sequence diagram of timetable generation

3.1.6 Database Design

Figure 3.8 refers to the database design of the proposed system.

3.2 User Interface Design

The main source of interaction between the user and the system is the user interface. Following are some GUI designs related to the proposed system.

3.2.1 Login

Users can log in to the system by using the login screen in Figure 3.9. Users should use a valid username and password to successfully log in.



Figure 3.9: User Interface of Login screen

3.2.2 Student Registration

Following Figure 3.10 refers to the student registration of the school. This UI is the use of the school administration office. Once the student registered on the school DB using the data entered on the following UI, the system will generate an Index no for the student.

CREATE STUDENT ADMISSION

Home > Student > Student > Create

Save



RETAKE

Admission Date *

Date of Birth *

Full Name *

Initials *

Last Name *

Emergency Contact Name *

Emergency Contact No *

Medium

Address *

Street Name

City *

Admitted Grade

Student Parents Details

Title	Name	Relationship	Occupation	Working Address	Office Telephone	Contact Mobile	Contact Home	Email	NIC No	+
-------	------	--------------	------------	-----------------	------------------	----------------	--------------	-------	--------	---

Student Siblings

Admission No	Name with initials	Relationship	+
--------------	--------------------	--------------	---

[Back to List](#)

Figure 3.10: User Interface of Student Registration

3.2.3 Teacher Registration

Figure 3.11 refers to the teacher registration of the school. Any teacher who is teaching at the school can be registered by the administration office of the school and once the registration completes only the teacher will enable to map into class as a class teacher or map into a timetable schedule as the teacher of the particular session.

CREATE STAFF MEMBER

Home Admin Staff Member Create

Save



Staff Number * 1856

Title * Mrs

Full Name * Nadeeka weerasinghe 1

Initials * N

Last Name * weerasinghe

RETAKE

Gender * Female

Address1 * 78/1

Address2

City * Colombo 02

Mobile No * 0715863259

Home Contact No

Status Active

Is Teacher

Joined Date

Emergency Contact Name *

Emergency Contact No * 0715896325

NIC No * 72569322V

School Email

Teacher Section Senior

Retired Date

[Back to List](#)

Figure 3.11: User Interface of Teacher Registration

3.2.4 Student Promotion

Figure 3.12 will be used by grade headteachers and the administrative staff of the college. Annual student promotions will be updated using the below UI. This UI is giving a facility to select students from the last year's grade (eg: If the user is selecting students to grade 8, this only shows students from last year's grade 7) including the class teacher of the class.

MANAGE CLASS PROMOTION

Home > Student > Class Promotion > Edit

Back to List

Year: 2022 Grade: Senior Section Grade 9

Promoting Criteria: Continue The Same Class **Drafted**

Save

Admission No	Name	Medium	Current Class	Promoting Class
29052	K D Udesh	Sinhala	9.A	▼
29053	B M K Suresh	Sinhala	9.A	▼

Figure 3.12: User Interface of Student Promotion creation

3.2.5 Student Extra Activities

Figure 3.13 refers to the extracurricular activity creation of students. One student can have several memberships and he can become a committee member in any year. Also, the student may have several achievements. The below interface is designed to allow several memberships and achievements for one student, and he may become a committee member in one day. Records of memberships are useful in generating leaving certificate of a student.

STUDENT EXTRA ACTIVITIES

Home > Student > Student Extra Activities > Index

Student: H A S Asmitha

Extra Activity Acheivements

Acheivement Name	Acheivement Description	AwardedDate	Remarks	+
------------------	-------------------------	-------------	---------	---

Extra Activity Positions

Position Name	FromDate	ToDate	Remarks	+
---------------	----------	--------	---------	---

Figure 3.13: User Interface of Club Membership Creation

3.3 Summary

Chapter 3 aims at the design architecture of the proposed system including its use cases, class diagram, ER diagram, Sequential diagrams, and UI designs.

4 CHAPTER - IMPLEMENTATION

This chapter describes the implementation methodology of the proposed system. This chapter includes the module interaction of the system, database, and major codes of the project used to develop the system. Also, the chapter describes the environment needed to implement the project for client use.

4.1 Related Technologies

This project for Nalanda College will be developed as a web-based application using ASP.Net MVC (Brainvire, 2021) technologies as shown in Figure 4.1. Since users of the system are connecting from several locations, this project requires a web-enabled technology with a centralized database server.

The front-end of the project will be implemented using HTML (Hypertext Markup Language), CSS, Bootstrap and JavaScript, jQuery as the scripting language while the ADO.Net technologies are used as the back end. MVC architecture is used to manage development easily while using visual studio 2019 combined with C# as the programming language. SQL Server 2019 technologies will be used to manage the database.

The unit testing will be performed during the development stage and system testing will be performed at the end of the development stage using manual testing methods. After completing the integration testing, UAT will be able to complete by the staff of the school, and the UAT document needs to be signed off by both parties.

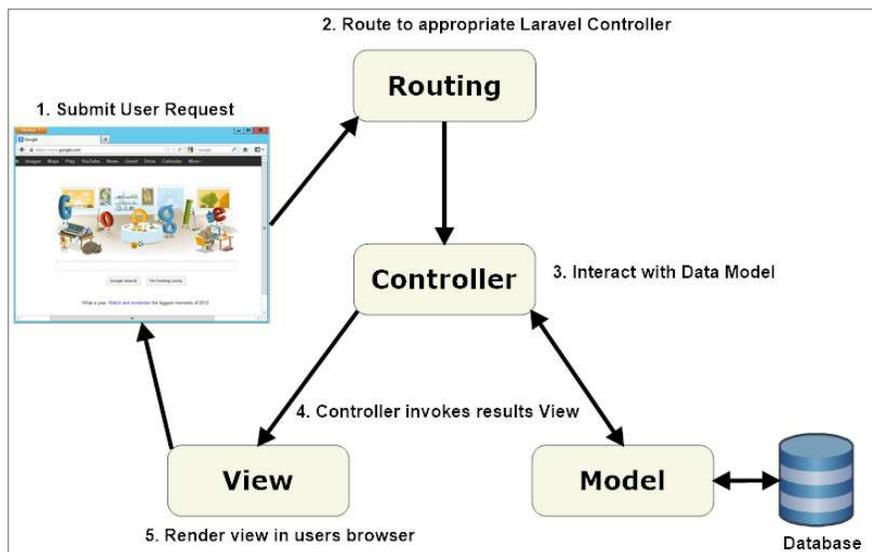


Figure 4.1: MVC Architecture

4.2 High-level Architecture of the System

The purpose of high-level architecture activities is to define a comprehensive solution based on principles, concepts, and properties logically related to and consistent with each other. The system architecture is abstract, conceptualization-oriented, global, and focused to achieve the mission and life cycle concepts of the system. It also focuses on the high-level structure in systems and system elements. The high-level architecture of the proposed system illustrates in Figure 4.2.

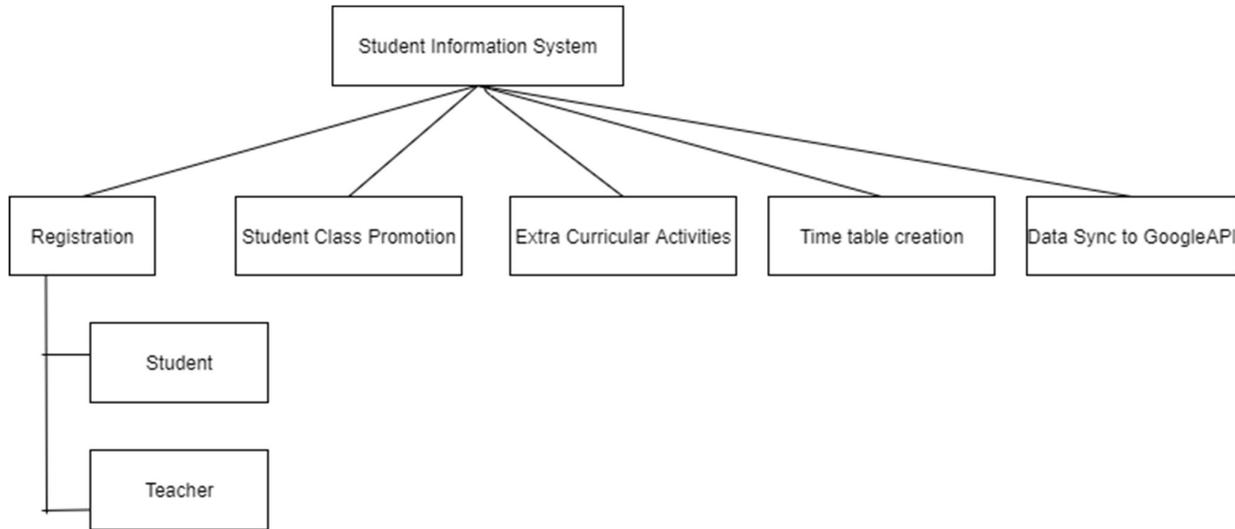


Figure 4.2: High-level Architecture

As the RAD (Rapid Application Development) (Nikolaieva, n.d.) methodology gives faster development and higher-quality development, system development is split into three phases. Application development was done phase-wise by using incremental and Iterative approach that emphasizes continuous user involvement.

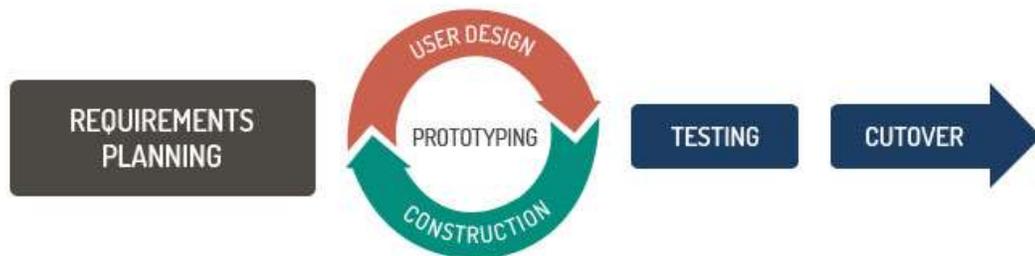


Figure 4.3: RAD Methodology

As the first phase student registration and teacher, registration was completed and as the second phase, the development of student class promotion and extra-curricular activity management was

implemented. As the third phase timetable management was implemented and as the final stage, student data, class promotions, and timetable data synchronization to Google API was implemented.

4.3 Modules of the system

The proposed system for Nalanda College is mainly categorized into the following four modules as in Figure 4.4.

- Admin Module
 - User Permission
 - User Creation
 - Staff Members
 - Visitors
 - Sections
 - Section Heads
 - Grades
 - Grade Heads
 - Extra Activities
- Academic Module
 - Subject Categories
 - Subjects
 - Grade Subjects
 - Grade Classes
 - Physical Classrooms
- Teacher Module
 - Teacher Information
 - Teacher Availability
- Student Module
 - Student Admission
 - Student Maintenance
 - Student Basket Subjects
 - Transfer Student
 - Student Marks
 - Class Promotion
 - Extra-curricular activities
- Online Module
 - Online Classroom
 - Online Timetable
- Google API Integration Module
- Report Generation Module

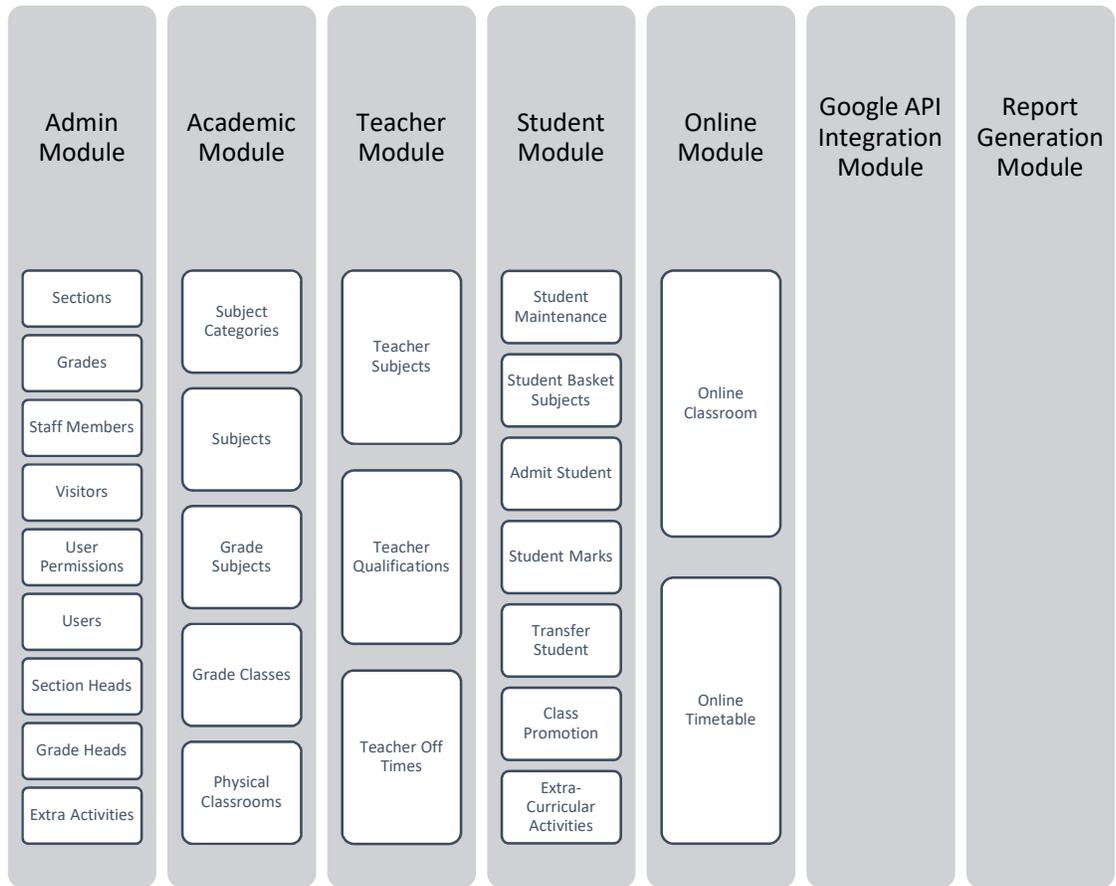


Figure 4.4: Module interaction of the system

The interoperability of the above modules is essential when the system accomplishes the required functionality.

4.4 Major Code Segment

Figure 4.5 refers to the file structure of the solution on visual studio. As the MVC design pattern is used to develop the system the file structure has three main folders as “Controllers”, “Models”, and “Views”.

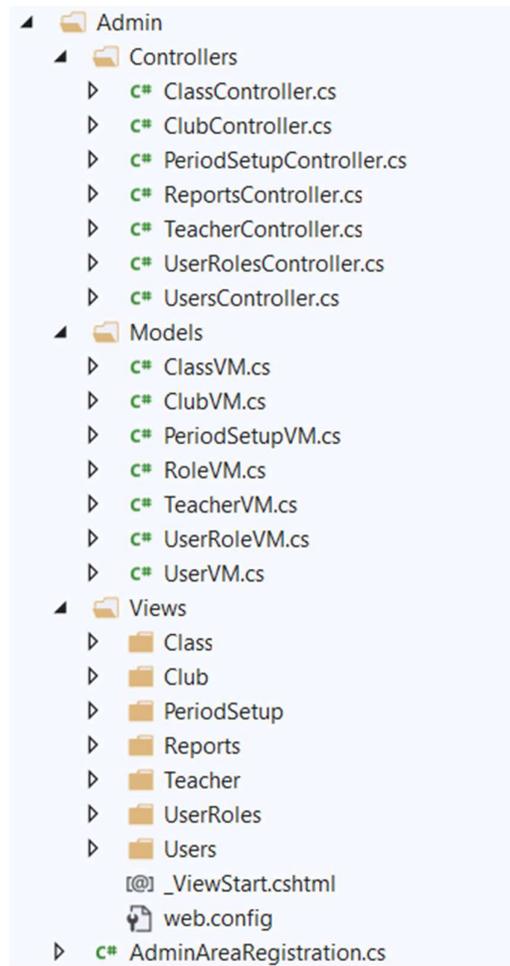


Figure 4.5: MVC file structure

4.4.1 HTML page sample

The following code in Figure 4.6 refers to the HTML page of a data entry UI of the “Teacher Information” of the project. This code will execute the create request UI of the project.

```

<div class="form-horizontal">
  <hr />
  <p>
    <input type="submit" value="Save" class="btn btn-primary" style="min-width:75px" />
  </p>
  <@Html.ValidationSummary(true, "", new { @class = "text-danger" })>
  <div class="form-group">
    <@Html.LabelFor(model => model.Title, htmlAttributes: new { @class = "control-label col-md-2" })>
    <div class="col-md-4">
      <@Html.EnumEditorFor(model => model.Title, htmlAttributes: new { @class = "form-control" })>
      <@Html.ValidationMessageFor(model => model.Title, "", new { @class = "text-danger" })>
    </div>
    <@Html.LabelFor(model => model.Gender, htmlAttributes: new { @class = "control-label col-md-2" })>
    <div class="col-md-4">
      <@Html.EnumEditorFor(model => model.Gender, htmlAttributes: new { @class = "form-control" })>
      <@Html.ValidationMessageFor(model => model.Gender, "", new { @class = "text-danger" })>
    </div>
  </div>

  <div class="form-group">...</div>
  <div class="form-group">...</div>
  <div class="form-group">
    <@Html.LabelFor(model => model.Address, htmlAttributes: new { @class = "control-label col-md-2" })>
    <div class="col-md-4">
      <@Html.EditorFor(model => model.Address, new { htmlAttributes = new { @class = "form-control" } })>
      <@Html.ValidationMessageFor(model => model.Address, "", new { @class = "text-danger" })>
    </div>
  </div>
  <div class="form-group">
    <@Html.LabelFor(model => model.ContactNo, htmlAttributes: new { @class = "control-label col-md-2" })>
    <div class="col-md-4">
      <@Html.EditorFor(model => model.ContactNo, new { htmlAttributes = new { @class = "form-control" } })>
      <@Html.ValidationMessageFor(model => model.ContactNo, "", new { @class = "text-danger" })>
    </div>
    <@Html.LabelFor(model => model.TelHome, htmlAttributes: new { @class = "control-label col-md-2" })>
    <div class="col-md-4">
      <@Html.EditorFor(model => model.TelHome, new { htmlAttributes = new { @class = "form-control" } })>
      <@Html.ValidationMessageFor(model => model.TelHome, "", new { @class = "text-danger" })>
    </div>
  </div>
  <div class="form-group">...</div>
  <br />
  <div class="form-group">...</div>
  <br />
  <div class="form-group">
    <@Html.LabelFor(model => model.Status, htmlAttributes: new { @class = "control-label col-md-2" })>
    <div class="col-md-4">
      <@Html.EnumEditorFor(model => model.Status, htmlAttributes: new { @class = "form-control" })>
      <@Html.ValidationMessageFor(model => model.Status, "", new { @class = "text-danger" })>
    </div>
    <@Html.LabelFor(model => model.InactiveReason, htmlAttributes: new { @class = "control-label col-md-2" })>
    <div class="col-md-4">
      <@Html.EditorFor(model => model.InactiveReason, new { htmlAttributes = new { @class = "form-control" } })>
      <@Html.ValidationMessageFor(model => model.InactiveReason, "", new { @class = "text-danger" })>
    </div>
  </div>
</div>
}

<br />
<div>
  <@Html.ActionLink("Back to List", "Index")>
</div>

@section Scripts {
  @Scripts.Render("~/Scripts/Admin/Teacher.js")
}

```

Figure 4.6: HTML code of Teacher Information

4.4.2 Js page sample

The following code segment in Figure 4.7 refers to the “Teacher.js” file of the project.

```
$.ajaxSetup({ cache: false });

$(function () {
    objFullName = $('#FullName');
    objStatus = $('#Status');
    objInactiveReason = $('#InactiveReason');

    objFullName.change(function () {
        var initials = "";
        var x = objFullName.val().split(' ');
        var a = x.length;
        $('#LName').val(x[a - 1].toString());

        for (var i = 0; i < a - 1; i++) { initials += x[i].charAt(0).toUpperCase() + " "; }

        $('#Initials').val(initials);
    });

    if (objStatus.val() == 0) {
        objInactiveReason.val("");
        SetComboReadOnly(objInactiveReason, true);
    }
    else {
        SetComboReadOnly(objInactiveReason, false);
    }

    objStatus.change(function () {
        if (objStatus.val() == 0) {
            objInactiveReason.val("");
            SetComboReadOnly(objInactiveReason, true);
        }
        else {
            SetComboReadOnly(objInactiveReason, false);
        }
    });
});
```

Figure 4.7: Code segment of a Js file

4.4.3 Save Student Enrollment

The following C# code segment in Figure 4.8 refers to the saved student enrollment of the project. This code is written in the controller of the MVC project.

```
public ActionResult Create([Bind(Include = "StudID,IndexNo,Title,Gender,FullName,Initials,LName,Address,DOB,!"  
"EmergencyConName,EmergencyContactTel,SpecialAttention,NameWithInt,ImagePath,Status,IsLeavingIssued,Inact  
{  
    try  
    {  
        var obj = (StudentVM)Session[sskCrtObj];  
        var existingStudent = db.Students.Where(e => e.FullName == student.FullName).FirstOrDefault();  
        if (existingStudent != null)  
        { ModelState.AddModelError("FullName", "Student Already Exist"); }  
  
        if (ModelState.IsValid)  
        {  
            student.CreatedBy = this.GetCurrUser();  
            student.CreatedDate = DateTime.Now;  
  
            var lastStudIndex = db.Students.MaxOrDefault(x => x.IndexNo);  
            student.IndexNo = lastStudIndex++;  
            var objStudent = db.Students.Add(student.GetEntity());  
            db.SaveChanges();  
  
            var imgPath = SaveImage(objStudent.StudID, student.ImagePath);  
            if (!imgPath.IsBlank())  
            {  
                objStudent.ImagePath = imgPath;  
                db.SaveChanges();  
            }  
  
            foreach (var det in obj.StudSublings)  
            {  
                det.SudID = objStudent.StudID;  
                det.CreatedBy = objStudent.CreatedBy;  
                det.CreatedDate = DateTime.Now;  
                db.StudSublings.Add(det.GetEntity());  
            }  
  
            foreach (var det in obj.StudFamilies)  
            {  
                det.StudID = objStudent.StudID;  
                det.CreatedBy = objStudent.CreatedBy;  
                det.CreatedDate = DateTime.Now;  
                db.StudFamilies.Add(det.GetEntity());  
            }  
  
            db.SaveChanges();  
  
            AddAlert(SMS.Common.AlertStyles.success, "Student Admission Created Successfully.");  
            return RedirectToAction("Details", new { id = objStudent.StudID });  
        }  
    }  
    catch (DbEntityValidationException dbEx)  
    { this.ShowEntityErrors(dbEx); }  
    catch (Exception ex)  
    { AddAlert(SMS.Common.AlertStyles.danger, ex.InnerException().Message); }  
  
    return View(student);  
}
```

Figure 4.8: Code segment - Save enrollment

4.4.4 Print PDF Report

The Following code segment in Figure 4.9 refers to the “Print Admission Card” function. This will out the admission card of the student once he gets registered with the system.

```
public ActionResult PrintAdmissionSheet(int id)
{
    var lstHdr = db.Students.Where(x => x.StudID == id).Select(x => new
    {
        x.StudID,
        Title = x.Title == TitleStud.Mr ? "Mr. " : "Ms.",
        AdmissionNo = x.IndexNo,
        FullName = x.FullName,
        Initials = x.Initials,
        LastName = x.LName,
        Address = x.Address,
        DOB = x.DOB,
        Gender = x.Gender == Gender.Female ? "Female" : "Male",
        EmergencyContactName = x.EmergencyConName,
        EmergencyContactTelno = x.EmergencyContactTel,
        SpecialAttention = x.SpecialAttention,
        DateRegistered = x.CreatedDate
    }).ToList();

    var lstSibDet = db.Students.Where(x => x.StudID == id).SelectMany(x => x.StudSiblings).Select(x => new
    {
        Name = x.StudentRelation.Title + ". " + x.StudentRelation.Initials + " " + x.StudentRelation.LName,
        Relationship = x.Relationship == SibRelationship.Brother ? "Brother" : "Sister",
        Class = x.StudentRelation.ClassStudents.Select(y => y.PromotionClass.Class.Grade).FirstOrDefault() + " - " + x.StudentID
    }).ToList();

    var lstFamDet = db.Students.Where(x => x.StudID == id).SelectMany(x => x.StudFamilies).Select(x => new
    {
        Name = x.Name,
        Relationship = x.Relationship == Relationship.Father ? "Father" : (x.Relationship == Relationship.Mother ? "Mother" :
        Occupation = x.Occupation,
        WorkingAdd = x.WorkingAdd,
        OfficeContact = x.OfficeTel,
        HomeContact = x.ContactHome,
        MobileContact = x.ContactMob,
        EmailAdd = x.Email,
        NICno = x.NICNo
    }).ToList();

    if (lstHdr.Count == 0)
    { return HttpNotFound(); }

    LocalReport report = new LocalReport();
    report.ReportPath = System.Web.HttpContext.Current.Server.MapPath("~/Reports/ApplicationForm.rdlc");

    ReportDataSource rds = new ReportDataSource();
    rds.Name = "dsStudent";
    rds.Value = lstHdr;
    report.DataSources.Add(rds);

    rds = new ReportDataSource();
    rds.Name = "dbSiblings";
    rds.Value = lstSibDet;
    report.DataSources.Add(rds);

    rds = new ReportDataSource();
    rds.Name = "dsFamily";
    rds.Value = lstFamDet;
    report.DataSources.Add(rds);

    Byte[] mybytes = report.Render("PDF");

    Response.AppendHeader("content-disposition", "inline; filename=file.pdf");
    return new FileStreamResult(new MemoryStream(mybytes), "application/pdf");
}
```

Figure 4.9: Code segment - Print PDF report

4.5 Implementation Environment

4.5.1 Hardware Environment

Following specification of hardware environment used to develop the proposed system.

- Operating System: Windows 10
- Processor: Intel CORE i7
- RAM: 8GB
- Hard disk: 1TB

4.5.2 Software and Technologies

The following list of tools and software technologies was used in the development phase of this system.

- Visual Studio 2019(.Net framework)
This IDE can use to edit, debug, and build code, and then publish an app. It is a feature-rich program that can be used for many aspects of software development. Over and above the standard editor and debugger that most IDEs provide, Visual Studio includes compilers, code completion tools, graphical designers, and many more features to ease the software development process.
- EF (Entity Framework)
Entity Framework is an open-source object-relational mapping (ORM) framework developed for ADO.NET by Microsoft. This is used in the application to automate all database-related activities such as connection open, data retrieval, create data set within the framework of .net.
As the application used EF, it used LINQ queries for the querying of DB.
- IIS 10 (Internet Information Systems)
IIS is used to host the application locally. This includes built-in authentication options such as Basic, ASP.NET, and Windows auth.
- SQL Server 2019
Microsoft SQL Server is a relational database management system developed by Microsoft. This is used to store and retrieve the data of the application.
- SQL Server Reporting Service

SSRS is a server-based report generating software system from Microsoft. This is used to produce formatted reports with tables in the form of data, graphs, images, and charts in the application.

- **HTML 5**
This is the markup language used for structuring and presenting the content of the web pages of the project.
- **CSS 3**
This is the style sheet language used to enable the separation of presentation and content, including layout, colours, and fonts of the application.
- **Bootstrap 3**
Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. This is used to develop forms of the application with responsive UI components such as buttons, navigations, divs and, etc.
- **jQuery 3.5.1**
jQuery is a lightweight JavaScript library that we can use in webpage development. Using jQuery makes it much easier to use JavaScript in the application.

4.6 Summary

This chapter refers to the implementation methodology of the proposed system. It includes the module interaction of the system, database design, and some of the major codes including the major operation of the project which used to develop the system.

5 CHAPTER – TESTING AND EVALUATION

This chapter elaborates on the developed system's evaluation procedure. Test plans, test cases, user acceptance test results were carried out to ensure that all the aspects of the system have been tested.

5.1 Related Testing Types

Testing of the project started at the development stage, and unit testing, integration testing, and system testing were carried out. User acceptance testing was conducted with the customer after completing the developer testing. (Padamkar, 2020)

5.1.1 Unit Testing

The individual component of each UI is tested to complete the unit testing of the project. This was completed during the development phase of the project by the developer. Unit testing is carried out during the development phase to identify the correctness of each component placed in each UI of the project. This was helped to find out errors of the individual components of each UI.

5.1.2 Integration Testing

An individual module that was subjected to unit testing was tested by integrating. A top-down approach was followed. Errors can occur at the integration was able to find at the end of this test.

5.1.3 System Testing

System testing is done in the fully completed system from the developer's end. The purpose of the system testing is to test the client's expected requirement is fulfilled by the developed system. This was done after the unit testing and integrated testing. Errors that were not detected in the previous phase were also exposed by system testing.

5.1.4 User Acceptance Testing

The secretary of the principal, the management assistant of the principal's office, some of the grade headteachers, some of the class teachers were helped with the acceptance testing, to make it a success.

Along with the functional testing methods, it was able to provide an error-free system for the Nalanda college, to prevent the obstacle which could occur latterly.

5.2 Test Cases

Following table 5.1 refers to some of the test cases tested in the project.

Test #	Test Description	Sample Input Values	Expected Outcome
1	Open the login screen and click on the login button	UserName and password are blank	An Alert should be shown saying UserName is required.
2	Open the login screen and click on the login button	With an invalid UserName or password	An Alert should be shown saying UserName or password is incorrect.
3	Open the login screen and click on the login button	With correct username and password	User should be redirected to the Home Page
4	Open the login screen and click on the login in with the Google button.	Login with a personal email	An alert should be shown saying logged in email account does not have access to the system
5	Open the login screen and click on the login in with the Google button.	Login with school email which has access to the system	User should be redirected to the Home Page
6	Click on the user permissions menu	None	The system should list all the defined user permission and should be able to filter.
7	Visit the create page and create a new user permission	Mandatory fields are not filled out	Error messages should appear under all the required fields
8	Visit the create page and create a new user permission	Enter an existing permission code	Error messages should appear saying permission code already exists.
9	Visit the create page and create a new user permission	Fill out all the details and select a few menu items	Should be able to save without an issue.
10	Select permission, visit the edit page and modify some info	Select some grades and menus	Should be able to save without an issue.
11	Select permission and delete it	None	The record should be deleted
12	Click on the user's menu	None	The system should list all the defined users and should be able to filter.
13	Visit the create page and create a new user	Mandatory fields are not filled out	Error messages should appear under all the required fields
14	Visit create the page and create a new user	Enter an existing username	Error messages should appear saying the username already exists.

15	Visit the create page and create a new user	Fill out all the details and select a few permissions	Should be able to save without an issue.
16	Select a user, visit the edit page, and modify some info	Add some permissions	Should be able to save without an issue.
17	Select a user and delete it	None	The record should be deleted
18	Click on the sections menu	None	The system should list all the defined sections and should be able to filter.
19	Visit the create page and create a new section	Mandatory fields are not filled out	Error messages should appear under all the required fields
20	Visit the create page and create a new section	Enter an existing section code	Error messages should appear saying section code already exists.
21	Visit the create page and create a new section	Fill out all the details	Should be able to save without an issue.
22	Select a section, visit the edit page, and modify some info	Change description	Should be able to save without an issue.
23	Select a section and delete it	None	The record should be deleted
24	Click on section heads menu	None	The system should list all the defined section heads and should be able to filter.
25	Visit the create page and create a new section head	Mandatory fields are not filled out	Error messages should appear under all the required fields
26	Visit the create page and create a new section head	Enter an existing section head	Error messages should appear saying section head already exists.
27	Visit the create page and create a new section head	Fill out with valid details	Should be able to save without an issue.
28	Select a section head, visit the edit page and modify some info	Change dates	Should be able to save without an issue.
29	Select a section head and delete it	None	The record should be deleted
30	Click on the grades menu	None	The system should list all the defined grades and should be able to filter.
31	Visit the create page and create a new grade	Mandatory fields are not filled out	Error messages should appear under all the required fields
32	Visit the create page and create a new grade	Enter an existing grade	Error messages should appear saying section code already exists.
33	Visit the create page and create a new grade	Fill out all the details	Should be able to save without an issue.
34	Select a grade, visit the edit page, and modify some info	Change description	Should be able to save without an issue.
35	Select a grade and delete it	None	The record should be deleted
36	Click on the grade heads menu	None	The system should list all the defined grade heads and should be able to filter.

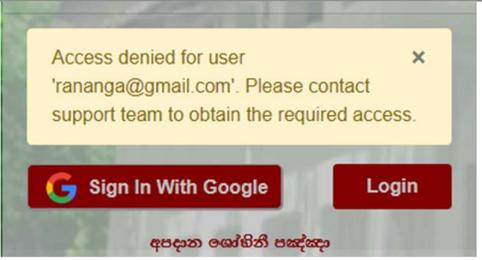
37	Visit the create page and create a new grade head	Mandatory fields are not filled out	Error messages should appear under all the required fields
38	Visit the create page and create a new grade head	Enter an existing grade head	Error messages should appear saying grade head already exists.
39	Visit the create page and create a new grade head	Fill out with valid details	Should be able to save without an issue.
40	Select a grade head, visit the edit page, and modify some info	Change dates	Should be able to save without an issue.
41	Select a grade head and delete it	None	The record should be deleted
42	Click on the extra activities menu	None	The system should list all the defined extra activities and should be able to filter.
43	Visit the create page and create a new extra activity	Mandatory fields are not filled out	Error messages should appear under all the required fields
44	Visit the create page and create a new extra activity	Enter an existing extra activity	Error messages should appear saying extra activity already exists.
45	Visit the create page and create a new extra activity	Fill out all the details	Should be able to save without an issue.
46	Select an extra activity, visit the edit page, and modify some info	Change positions & achievements	Should be able to save without an issue.
47	Select an extra activity and delete it	None	The record should be deleted

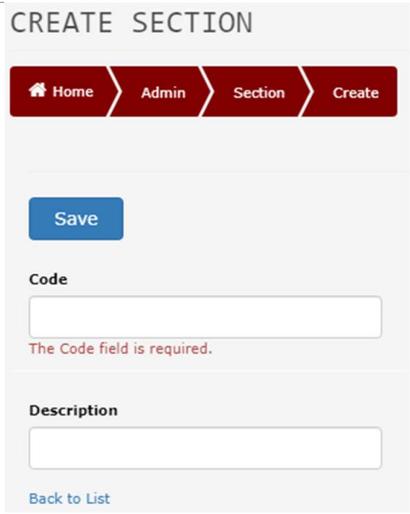
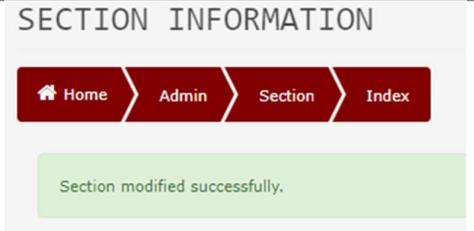
Table 5.1: Test Cases

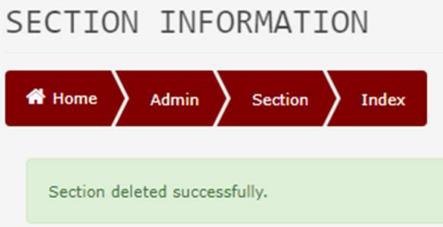
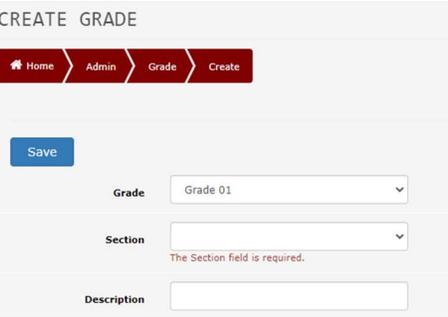
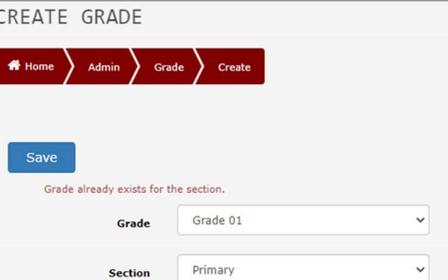
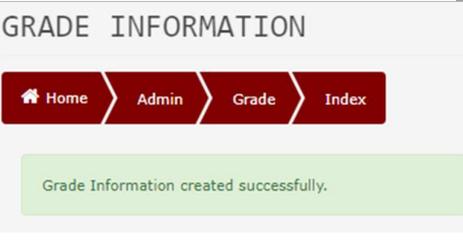
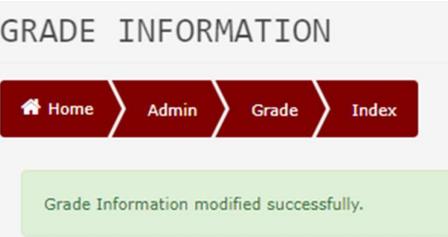
5.3 Test Results

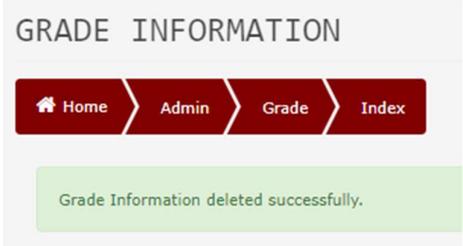
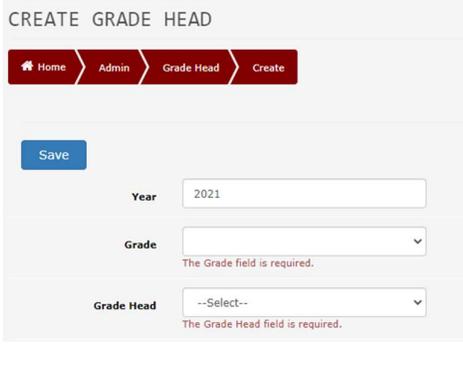
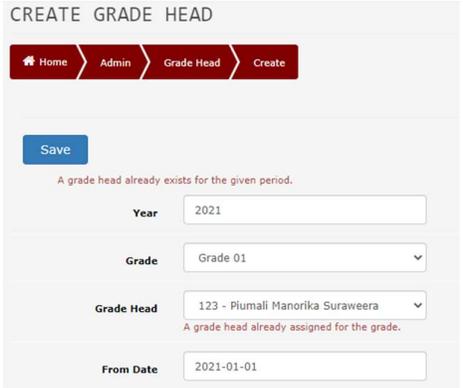
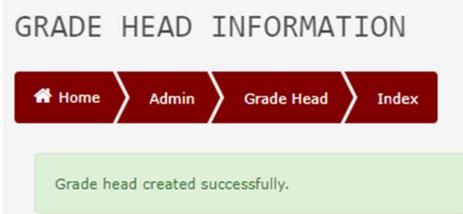
Table 5.2 refers to the actual test results of the above test cases. Additional test results of the developed system are attached in Appendix D.

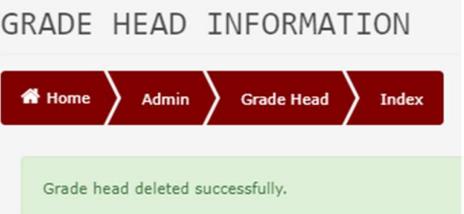
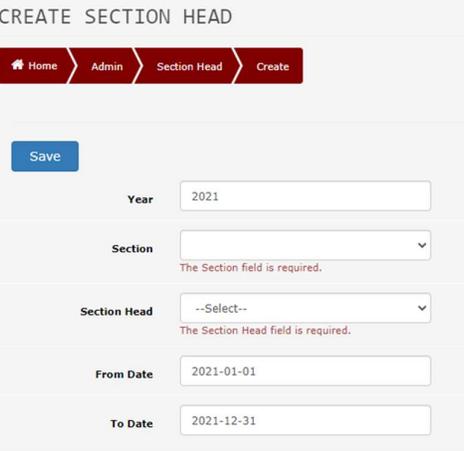
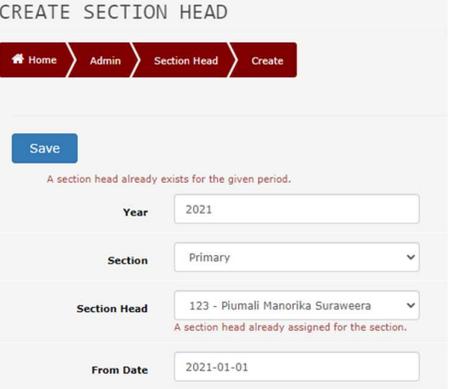
Test #	Test Description	Input Values	Expected Outcome	Result
Login				
1	Open the login screen and click on the login button	UserName and password are blank	An Alert should be shown saying UserName is required.	

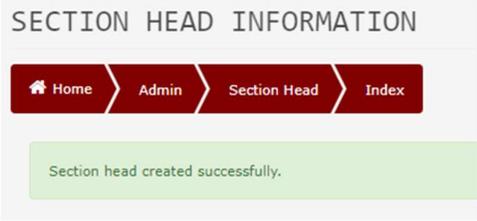
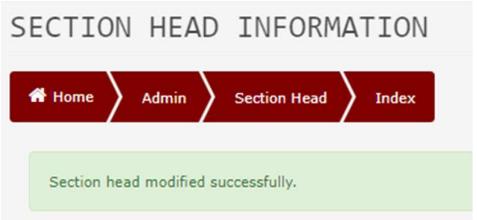
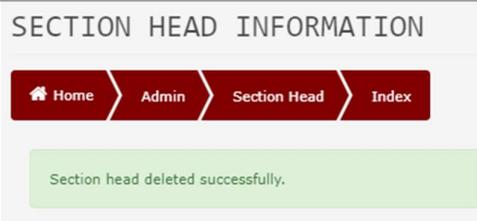
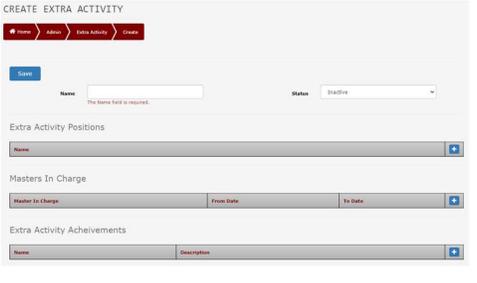
2	Open the login screen and click on the login button	With an invalid UserName or password	An Alert should be shown saying UserName or password is incorrect.	
3	Open the login screen and click on the login button	With correct username and password	User should be redirected to the Home Page	
4	Open the login screen and click on the login in with the Google button.	Login with a personal email	An alert should be shown saying logged in email account does not have access to the system	
5	Open the login screen and click on the login in with the Google button.	Login with school email which has access to the system	User should be redirected to the Home Page	
Sections				

6	Click on the sections menu	None	The system should list all the defined sections and should be able to filter.	
7	Visit the create page and create a new section	Mandatory fields are not filled out	Error messages should appear under all the required fields	
8	Visit the create page and create a new section	Enter an existing section code	Error messages should appear saying section code already exists.	
9	Visit the create page and create a new section	Fill out all the details	Should be able to save without an issue.	
10	Select a section, visit the edit page, and modify some info	Change description	Should be able to save without an issue.	

11	Select a section and delete it	None	The record should be deleted	
Grades				
12	Click on the grades menu	None	The system should list all the defined grades and should be able to filter.	
13	Visit the create page and create a new grade	Mandatory fields are not filled out	Error messages should appear under all the required fields	
14	Visit the create page and create a new grade	Enter an existing grade for the section	Error messages should appear saying a grade already exists for the section.	
15	Visit the create page and create a new grade	Fill out all the details	Should be able to save without an issue.	
16	Select a grade, visit the edit page, and modify some info	Change description	Should be able to save without an issue.	

17	Select a grade and delete it	None	The record should be deleted	
Grade Heads				
18	Click on the grade heads menu	None	The system should list all the defined grade heads and should be able to filter.	Grade heads are loaded and can be filtered. Pass
19	Visit the create page and create a new grade head	Mandatory fields are not filled out	Error messages should appear under all the required fields	
20	Visit the create page and create a new grade head	Enter an existing grade head for a given period	Error messages should appear saying grade head already assigned.	
21	Visit the create page and create a new grade head	Fill out with valid details	Should be able to save without an issue.	

22	Select a grade head, visit the edit page, and modify some info	Change dates	Should be able to save without an issue.	
23	Select a grade head and delete it	None	The record should be deleted	
Section Heads				
24	Click on section heads menu	None	The system should list all the defined section heads and should be able to filter.	
25	Visit the create page and create a new section head	Mandatory fields are not filled out	Error messages should appear under all the required fields	
26	Visit the create page and create a new section head	Enter an existing grade head for a given period	Error messages should appear saying section head already assigned.	

27	Visit the create page and create a new section head	Fill out with valid details	Should be able to save without an issue.	
28	Select a section head, visit the edit page and modify some info	Change dates	Should be able to save without an issue.	
29	Select a section head and delete it	None	The record should be deleted	
Extra Activities				
30	Click on the extra activities menu	None	The system should list all the defined extra activities and should be able to filter.	
31	Visit the create page and create a new extra activity	Mandatory fields are not filled out	Error messages should appear under all the required fields	
32	Visit the create page and create a new extra activity	Enter an existing extra activity	Error messages should appear saying extra activity already exists.	

33	Visit the create page and create a new extra activity	Fill out all the details	Should be able to save without an issue.	<p>EXTRA ACTIVITY INFORMATION</p> <p>Home Admin Extra Activity Index</p> <p>Extra Activity Created Successfully.</p>																														
34	Select an extra activity, visit the edit page, and modify some info	Change positions & achievements	Should be able to save without an issue.	<p>EXTRA ACTIVITY INFORMATION</p> <p>Home Admin Extra Activity Index</p> <p>Extra Activity Modified Successfully.</p>																														
35	Select an extra activity and delete it	None	The record should be deleted	<p>EXTRA ACTIVITY INFORMATION</p> <p>Home Admin Extra Activity Index</p> <p>Extra Activity Deleted Successfully.</p>																														
Class Promotions																																		
36	Click on the Student -> class promotion menu	None	The system should list all the defined class promotions and should be able to filter.	<p>CLASS PROMOTIONS</p> <p>Home Student Class Promotion Index</p> <p>Create</p> <p>Search By Year <input type="text"/> Search</p> <table border="1"> <thead> <tr> <th>Year ID</th> <th>Grade</th> <th>Promoting Criteria</th> <th>Status</th> </tr> </thead> <tbody> <tr> <td>2022</td> <td>Primary Section Grade 1</td> <td>Continue The Same Class</td> <td>Finalized</td> </tr> </tbody> </table> <p>10 Rows Per Page</p>	Year ID	Grade	Promoting Criteria	Status	2022	Primary Section Grade 1	Continue The Same Class	Finalized																						
Year ID	Grade	Promoting Criteria	Status																															
2022	Primary Section Grade 1	Continue The Same Class	Finalized																															
37	Visit the create page and start processing a new class promotion	From grade is not entered	An error message should appear under the field	<p>CREATE CLASS PROMOTION</p> <p>Home Student Class Promotion Create</p> <p>Process</p> <p>Year 2022 From Grade <input type="text"/> The From Grade field is required.</p> <p>Promoting Criteria Continue The Same Class</p> <p>Back to List</p>																														
38	Visit the create page and start processing a new class promotion	Enter an existing year and from grade	Error messages should appear saying a class promotion is already exists.	<p>CREATE CLASS PROMOTION</p> <p>Home Student Class Promotion Create</p> <p>Process</p> <p>A class promotion already exists for the selected year and grade.</p> <p>Year <input type="text" value="2022"/></p>																														
39	Visit the create page and start processing a new class promotion	Enter a valid year and a from grade	Class promotion is created and students are displayed in edit mode with their promoting class	<p>MANAGE CLASS PROMOTION</p> <p>Home Student Class Promotion Edit</p> <p>Back to List</p> <p>Year 2022 Grade Primary Section Grade 1</p> <p>Promoting Criteria Continue The Same Class Drafted</p> <p>Save</p> <table border="1"> <thead> <tr> <th>Admission No</th> <th>Name</th> <th>Medium</th> <th>Current Class</th> <th>Promoting Class</th> </tr> </thead> <tbody> <tr> <td>28951</td> <td>L O T S Samaneworpa</td> <td>Orthals</td> <td>1.A</td> <td>2.A</td> </tr> <tr> <td>28956</td> <td>M A S Amotika</td> <td>Orthals</td> <td>1.A</td> <td>2.A</td> </tr> <tr> <td>28951</td> <td>B V D A A Parwa</td> <td>Orthals</td> <td>1.A</td> <td>2.A</td> </tr> <tr> <td>28958</td> <td>T V Ametia</td> <td>Orthals</td> <td>1.A</td> <td>2.A</td> </tr> <tr> <td>18411</td> <td>M I K Poluha</td> <td>Kothals</td> <td>1.A</td> <td>2.A</td> </tr> </tbody> </table>	Admission No	Name	Medium	Current Class	Promoting Class	28951	L O T S Samaneworpa	Orthals	1.A	2.A	28956	M A S Amotika	Orthals	1.A	2.A	28951	B V D A A Parwa	Orthals	1.A	2.A	28958	T V Ametia	Orthals	1.A	2.A	18411	M I K Poluha	Kothals	1.A	2.A
Admission No	Name	Medium	Current Class	Promoting Class																														
28951	L O T S Samaneworpa	Orthals	1.A	2.A																														
28956	M A S Amotika	Orthals	1.A	2.A																														
28951	B V D A A Parwa	Orthals	1.A	2.A																														
28958	T V Ametia	Orthals	1.A	2.A																														
18411	M I K Poluha	Kothals	1.A	2.A																														

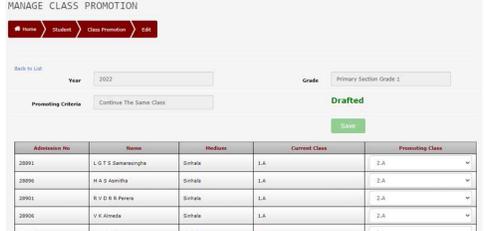
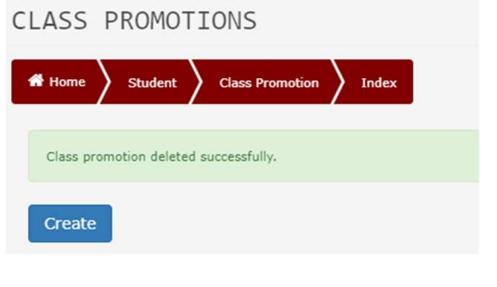
40	Visit the class promotions index page and select a drafted class promotion	Go to Edit mode	Eligible Students are displayed and are in edit mode	
41	Visit the class promotions index page and select a drafted class promotion	Delete the class promotion	Should be able to delete and a message should be displayed	
42	Visit the class promotions index page and select a drafted class promotion	Go to Edit mode and change promoting class for a student	Changes should be saved instantly	
43	Visit the class promotions index page and select a drafted class promotion	Click on Finalize	Confirmation dialogue should be displayed	
44	Visit the class promotions index page and select a drafted class promotion	Click on Finalize and confirm	The success message is displayed and status updated	
45	Visit the class promotions index page and select a finalized class promotion	None	Student details are displayed in read-only mode	

Table 5.2: Test Results

5.4 User Evaluation

User evaluation was completed using google forms among users of the system. The evaluation form that was shared with the school administration is attached to figure 5.1 and 5.2. The evaluation form was shared with 15 users and only 13 responded during the period. User-level participation for the survey is as follows.

- An office assistant for the Acting Principal – 1 User
- Grade headteachers – 3 Users
- Class teachers – 5 Users
- IT teachers – 3 Users
- Office assistant – 3 Users

Survey - Student Management System for Nalanda College

* Required

1. How well does the product meet your needs? *

Mark only one oval.

- Extremely well
- Very well
- Somewhat well
- Not so well

2. How easy is it to navigate the application? *

Mark only one oval.

- Very easy
- Somewhat easy
- Not so easy
- Not easy at all

3. How satisfied are you with this product's ease of use? *

Mark only one oval.

- Extremely Satisfied
- Very Satisfied
- Somewhat Satisfied
- Not so satisfied
- Not at all satisfied

Figure 5.1: Survey - Google Form page1

4. How satisfied are you with the look and feel of the application? *

Mark only one oval.

- Extremely Satisfied
- Very Satisfied
- Somewhat Satisfied
- Not so satisfied
- Not at all satisfied

5. How satisfied are you with this product's performance? *

Mark only one oval.

- Extremely Satisfied
- Very Satisfied
- Somewhat Satisfied
- Not so satisfied
- Not at all satisfied

6. Which of the following would you use to describe the product? *

Check all that apply.

- Reliable
- High Quality
- Useful
- Unique
- Impractical
- Ineffective
- Poor Quality
- Unreliable

7. How likely are you to recommend the product to a friend or colleague? *

Mark only one oval.

	1	2	3	4	5	
Not Like	<input type="radio"/>	Very Like				

8. Do you have any other comment or concern?

This content is neither created nor endorsed by Google.

Google Forms

Figure 5.2: Survey - Google Form page2

5.4.1 Results of the Testing

Figure 5.3 and 5.4 refers to feedback responses of the survey.

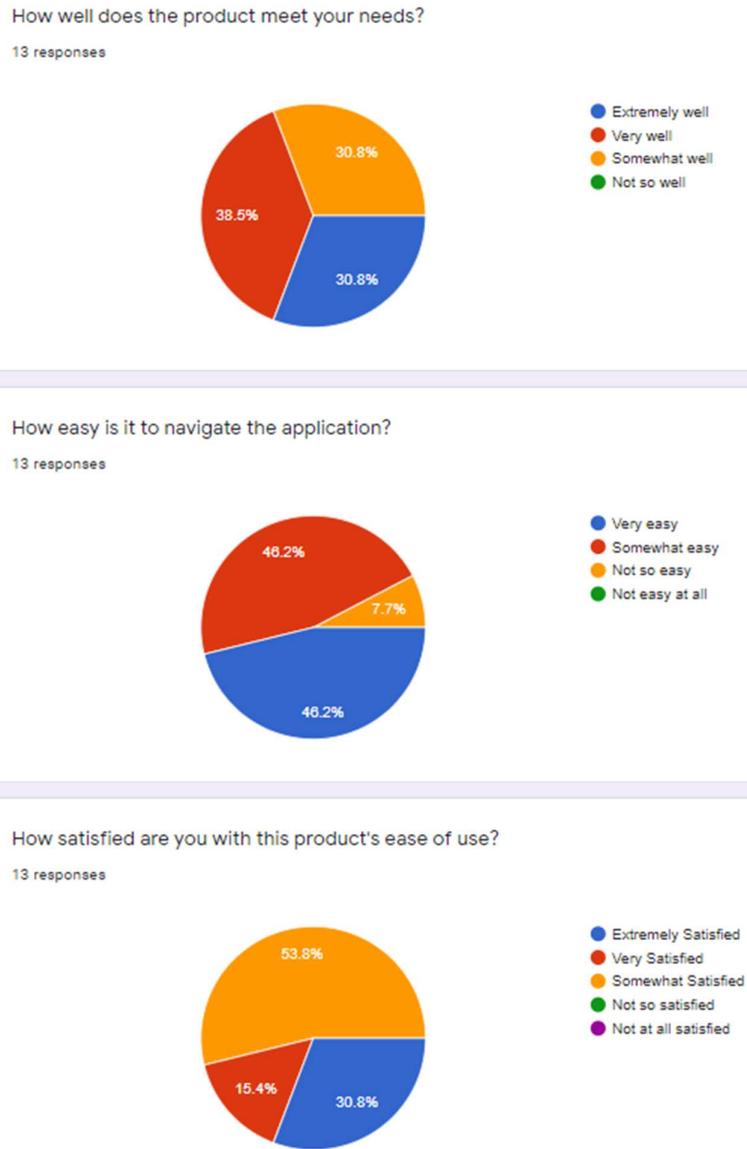
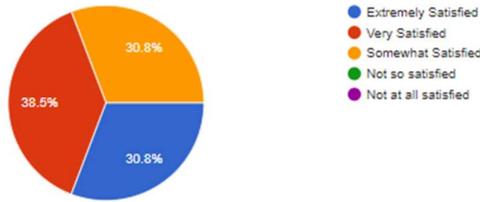


Figure 5.3: Survey Responses Q1-3

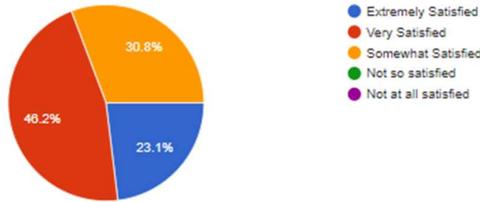
How satisfied are you with the look and feel of the application?

13 responses



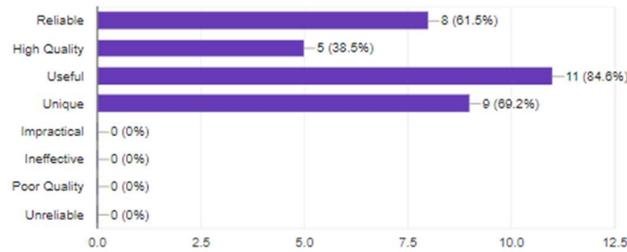
How satisfied are you with this product's performance?

13 responses



Which of the following would you use to describe the product?

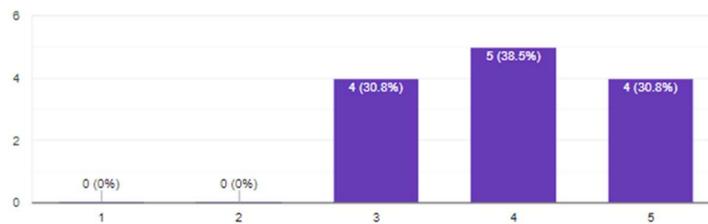
13 responses



How likely are you to recommend the product to a friend or colleague?



13 responses



Do you have any other comment or concern?

4 responses

Attendance reports are incorrect some times. Please fix them

Very useful app. Thank you

Overall a good application, But some screens should be improved to be more user friendly.

Good job, extremely satisfied.

Figure 5.4: Survey Responses Q4-8

5.5 Summary

This chapter discussed the user evaluation process of the entire developed system as it is an essential process to proceed to get actual user feedback regarding their experience with the system.

The next chapter will provide a detailed description regarding lessons learned, the future work, and a brief on the entire project.

6 CHAPTER – CONCLUSION

A system implementation for student management and online timetable management was discussed by the principal of Nalanda College during the COVID 19 pandemic period, not to affect to school process and make the school process streamline with an online platform. As the school decided to use Google classroom as an online LMS, the implementing system needed to synchronize with the Google classroom.

The system was needed to introduce mainly to increase the efficiency of the school administration by reducing their manual work. Also, they were looking for a solution for student index no duplication as well.

The online timetable and online student attendance also need to be managed through google APIs as per the school requirement. The school management system needed to be managed the student from their admission to their graduation till the leaving certificate was generated as per the requirement.

.Net technologies were selected to use in the development stage to fulfil the requirement of the project by considering its hosting and features. School had owned an azure cloud hosting feature and it was managed by another team of the school. As .Net was selected for development, SQL server management was selected as the DBMS tool of the project.

The requirements were gathered by having several interviews with several kinds of users of the application at the requirement gathering stage when the project was initiated. The project used the incremental waterfall method as the SDLC methodology. After completion of the implementation, UAT was done by the school administrative offices. Also, to collect their feedback, an online survey was conducted and according to its comments the system was at a satisfactory level and it needed much more improvements as well.

6.1 Lessons Learnt

This was a great opportunity for me to apply the previously learned lessons to a practical environment. As the domain area was familiar for me, therefore it helped to gather business requirements easily and it broadened my horizons into understanding, how to map related business processes into a computerized system.

In addition, this gave me an exceptional experience of being in a complete software development life cycle, starting from feasibility studies to the conclusion of the project. This project allowed getting extensive knowledge on several technical tools of every phase of the SDLC.

6.2 Future Improvement

The below-listed features are planned to add in the future to the newly built system as further improvements.

- School physical timetable management
- Parental portal to get their child's academic information and notifications
- School interview process of grade 1 admissions

7 REFERENCES

- 1) AltexSoft, 2019. *Non-functional Requirements: Examples, Types, How to Approach*. [Online]
Available at: <https://www.altexsoft.com/blog/non-functional-requirements/>
[Accessed 06 2021].
- 2) Anon., n.d. *open-source-school-management-solutions*. [Online]
Available at: <https://medevel.com/7-open-source-school-management-solutions/>
- 3) Anon., n.d. *school-software*. [Online]
Available at: <https://fekara.com/>
- 4) Brainvire, 2021. *Six Benefits Of Using MVC Model For Effective Web Application Development*. s.l.:Brainvire.
- 5) Etherington, D., 2017. *TechCrunch*. [Online]
Available at: <https://techcrunch.com/2017/04/27/google-classroom-now-lets-anyone-school-anyone-else/>
- 6) fekara.com, 2020. *Fekara*. [Online]
Available at: <https://fekara.com/school-software/>
[Accessed 12 February 2021].
- 7) Javier Soltero, V. P. & G. G. S., 2020. *Google.com*. [Online]
Available at: <https://blog.google/products/meet/bringing-google-meet-to-more-people/>
- 8) Nikolaev, A., n.d. *Up Tech - 8 Best Software Development Methodologies*. [Online]
Available at: <https://uptech.team/blog/software-development-methodologies>
[Accessed 2021].
- 9) Pedamkar, P., 2020. *educba.com*. [Online]
Available at: <https://www.educba.com/system-testing/>
[Accessed 06 2021].

APPENDIX A – SYSTEM DOCUMENTATION

This documentation is to help the system administrators who wish to implement and enhance this system, to install and configure the system from a technical perspective. The system documentation can be referred to if there are any changes to be made in the Daily Tasks Management System. To develop the system locally below software is required.

- Visual Studio 2019 Community Edition
- SQL Server Express 2019

The device chosen to install the above software should meet the following minimum requirements of hardware and software listed on A.1.

<p>Supported Operating Systems</p>	<p>Visual Studio 2019 & SQL Server 2019 will install and run on the following operating systems (64 bit recommended; ARM is not supported):</p> <ul style="list-style-type: none"> • Windows 10 version 1703 or higher: Home, Professional, Education, and Enterprise (LTSC and S are not supported) • Windows Server 2019: Standard and Datacenter • Windows Server 2016: Standard and Datacenter • Windows 8.1 (with Update 2919355): Core, Professional, and Enterprise • Windows Server 2012 R2 (with Update 2919355): Essentials, Standard, Datacenter • Windows 7 SP1 (with latest Windows Updates): Home Premium, Professional, Enterprise, Ultimate
<p>Hardware</p>	<ul style="list-style-type: none"> • 1.8 GHz or faster processor. Quad-core or better recommended • 2 GB of RAM; 8 GB of RAM recommended (2.5 GB minimum if running on a virtual machine) • Hard disk space: Minimum of 800MB up to 210 GB of available space, depending on features installed; typical installations require 20-50 GB of free space. • Hard disk speed: to improve performance, install Windows and Visual Studio on a solid-state drive (SSD).

	<ul style="list-style-type: none"> • A video card that supports a minimum display resolution of 720p (1280 by 720); Visual Studio will work best at a resolution of WXGA (1366 by 768) or higher.
Additional Requirements	<ul style="list-style-type: none"> • Administrator rights are required to install Visual Studio. • .NET Framework 4.5.2 or above is required to install Visual Studio. Visual Studio requires .NET Framework 4.7.2 to run, and this will be installed during setup. • Internet Explorer 11 or Edge is required for internet-related scenarios. Some features might not work unless these, or a later version, are installed. • Running Visual Studio 2019 (Professional, Community, and Enterprise) in Windows containers is not supported.

A. 1: Minimum Requirements

Installing SQL Server Express 2019

The steps to install SQL server express 2019 is as follows.

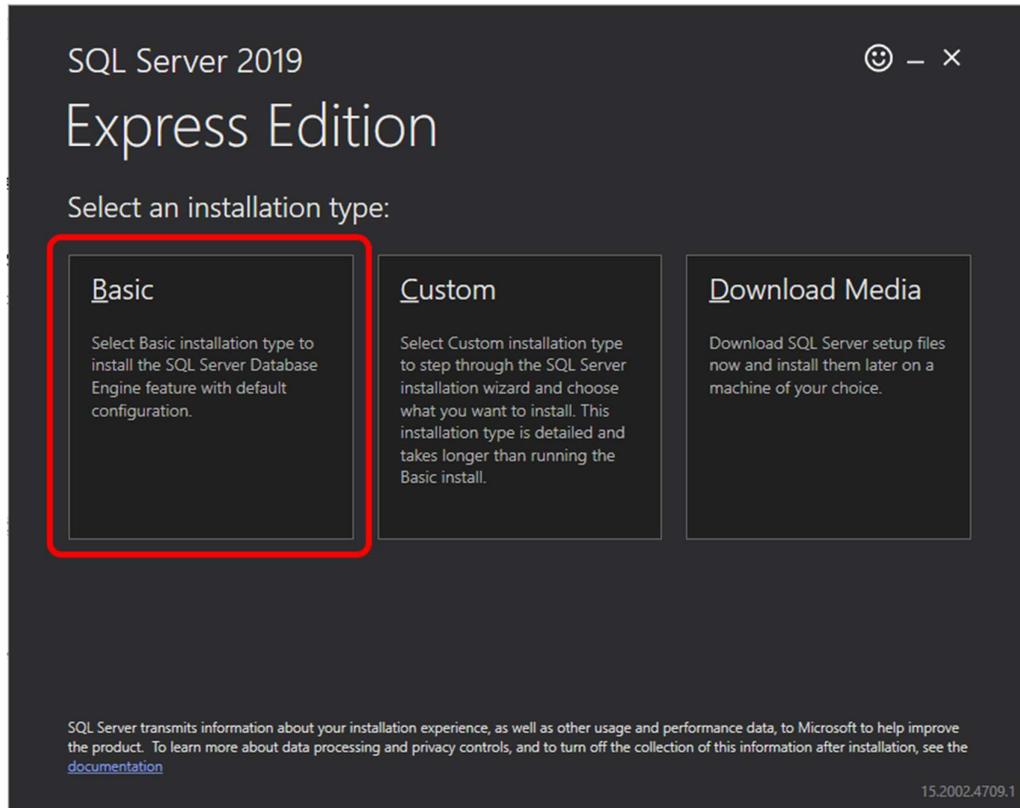
1. Download the SQL Server setup from <https://www.microsoft.com/en-us/sql-server/sql-server-downloads>

Also, make sure to select the correct edition when downloading. Express edition is free and suitable for this system as shown in Figure A2.



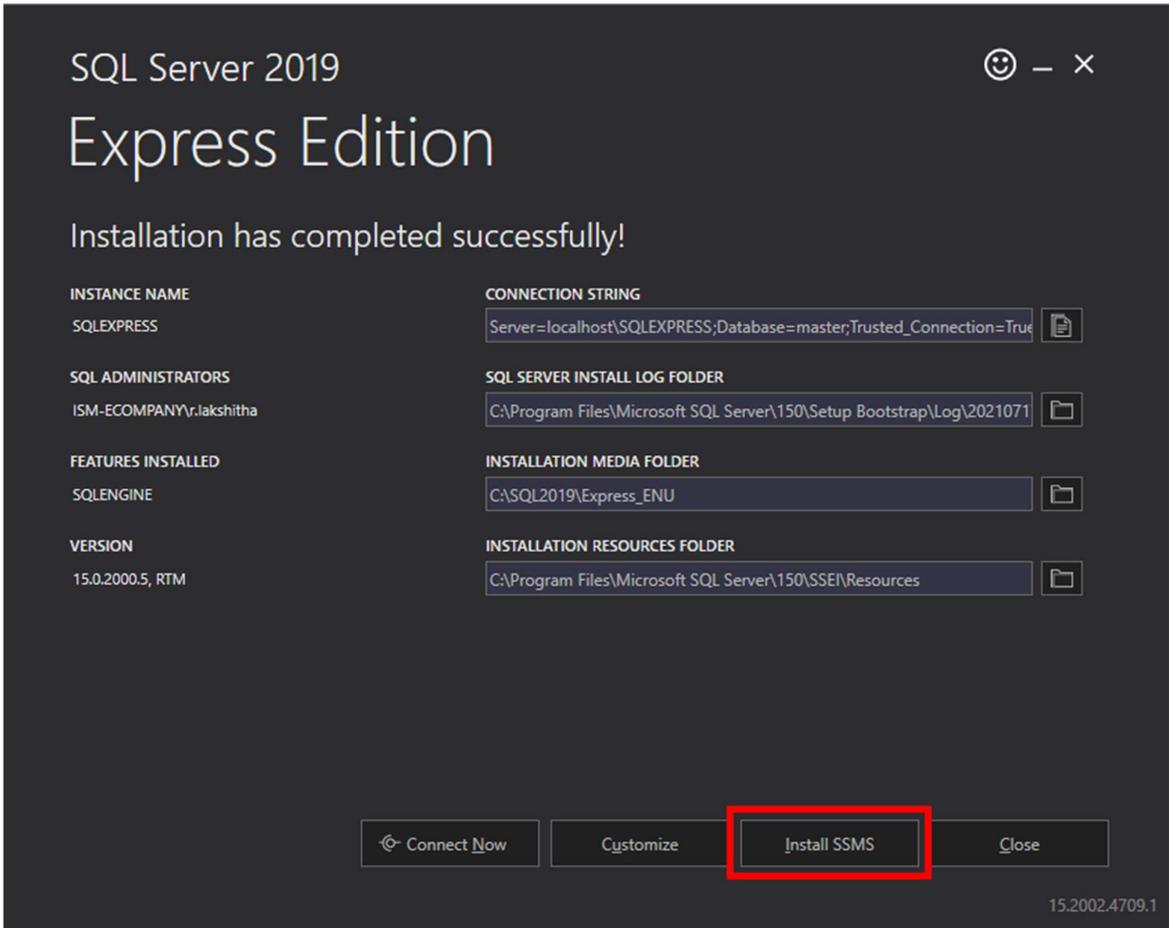
A. 2: SQL Installation-1

2. To start the installation process, double-click on the downloaded setup. Administrator rights are required to install.
3. Then the below screen will appear. Select “Basic” installation type. Refer Figure A.3.



A. 3: SQL Installation-2

4. Specify the installation folder, choose your own or keep the default folder suggested by the SQL Server installer.
5. Once the installation is completed, the below screen will be displayed. Please note down the connection string which will be required later. Now click on “Install SSMS” to install the SQL Server Management Studio which is useful for DB operations like querying data.

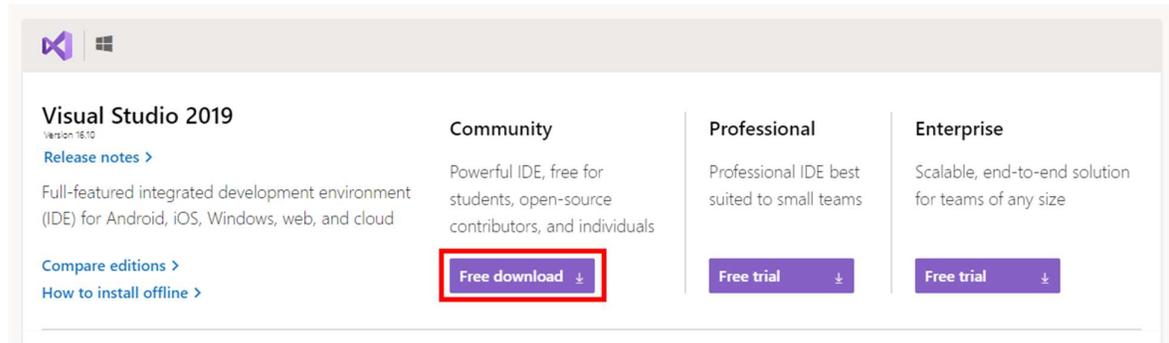


A. 4: SQL Installation-3

Install Visual Studio 2019

The steps to install Visual Studio 2019 is as follows.

1. Download the Visual Studio setup from <https://visualstudio.microsoft.com/downloads/>. Please make sure to select the correct edition when downloading. The Community edition is free and suitable for this system.



A. 5: VS Installation-1

2. To start the installation process, double-click on the downloaded setup. Administrator rights are required to install. This will download and start the visual studio installer.

Visual Studio Installer

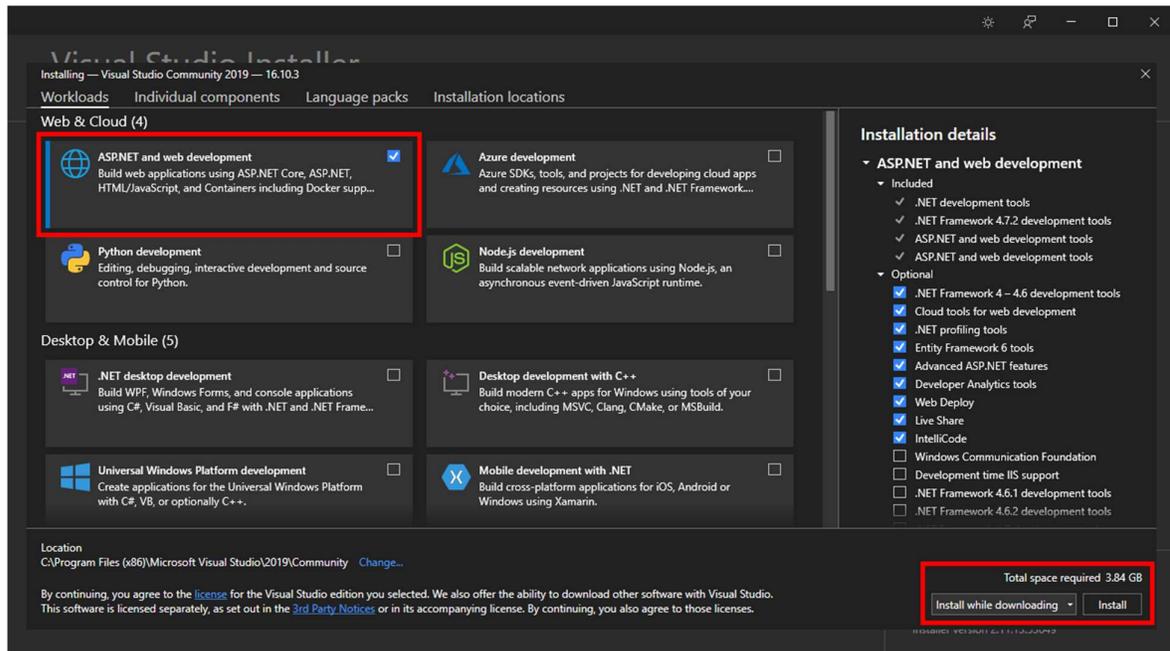
Just a moment ... Fetching your files.



Cancel

A. 6: VS Installation-2

- Once installed, Visual Studio Installer will be started automatically. Please select the below-highlighted options and continue with the installation.

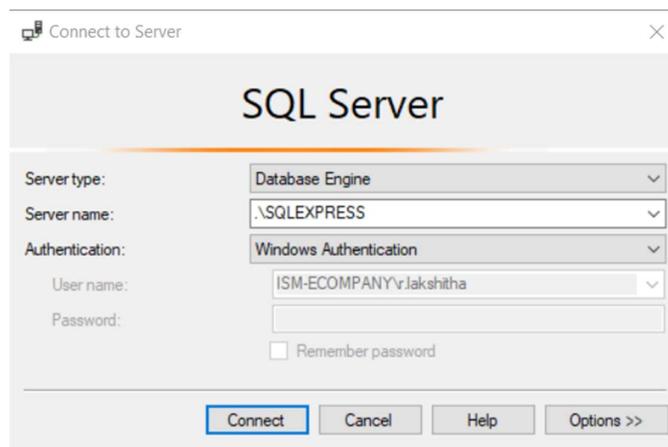


A. 7: VS Installation-3

Setup the Student Information System

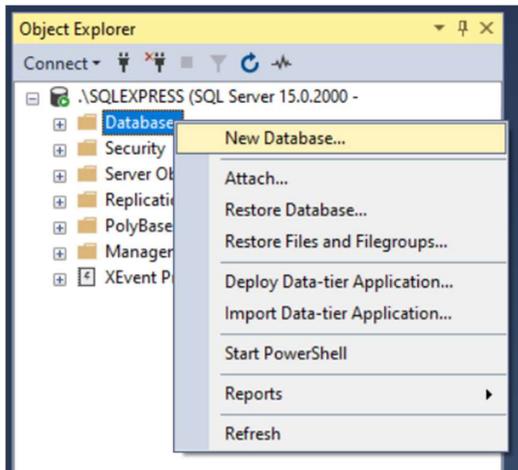
The steps to set up the Student Information System are as follows.

- Open SQL Server Management Studio and login using Windows Authentication



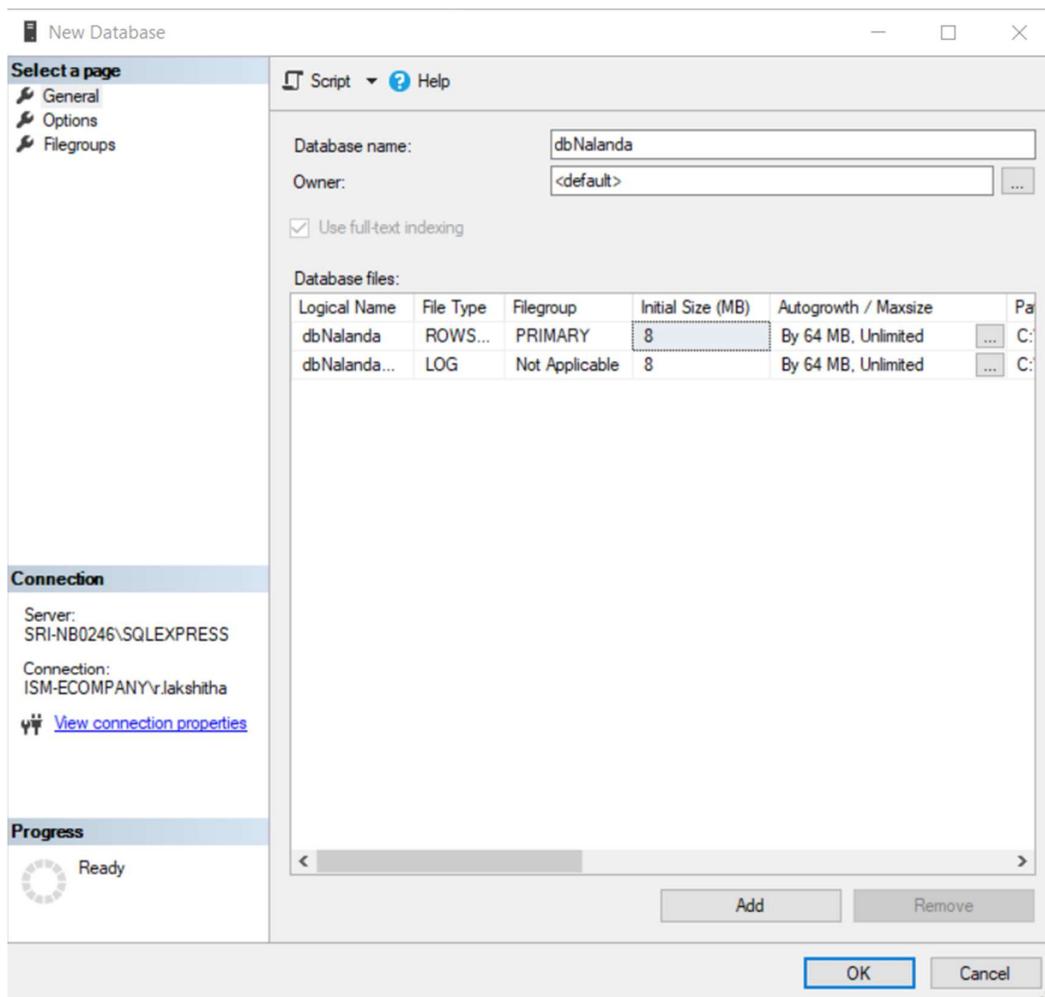
A. 8: Setup-1

- In Object Explorer, right-click on Databases and select “New Database”.



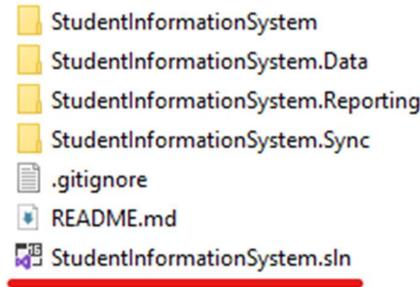
A. 9: Setup-2

- Provide a name for the Database and click ok. Please note down the given name.



A. 10: Setup-3

4. Extract the source code to a folder and open the “StudentInformationSystem.sln” file from Visual Studio.



A. 11: Setup-4

5. Open solution explorer and locate the “dbNalandaContext.cs” file in “StudentInformationSystem.Data” project. Then set the ConnectionString which you noted down when installing the SQL Server. Then change the database name of that connection string from “master” to the given DB name in step 3.

- E.g.

```
Server=localhost\SQLEXPRESS;Database=dbNalanda;Trusted_
Connection=True;
```

```

11 | 51 references | Rananga Suraweera, 24 days ago | 1 author, 4 changes
12 | public partial class dbNalandaContext : DbContext
13 | {
14 |     2 references | Rananga Suraweera, 27 days ago | 1 author, 1 change
15 |     public static string ConnectionString { get; set; }
16 |
17 |     16 references | Rananga Suraweera, 27 days ago | 1 author, 1 change
18 |     public dbNalandaContext()
19 |     {
20 |         ConnectionString = "Server=localhost\SQLEXPRESS;Database=dbNalanda;Trusted_Connection=True;";
21 |     }

```

A. 12: Setup-5

6. In Visual Studio open “Package Manager Console” by following the menu path Tools > Nuget Package Manager > Package Manager Console
7. In “Package Manager Console”, set the default project to StudentInformationSystem.Data. Then run the below command to update the database.
 - Update-Database

```
Package Manager Console
Package source: All [gear icon] Default project: StudentInformationSystem.Data [dropdown icon]
Each package is licensed to you by its owner. NuGet is not responsible for, nor does it grant any
to determine any dependencies.

Package Manager Console Host Version 5.10.0.7240

Type 'get-help NuGet' to see all available NuGet commands.

PM> Update-Database ←
Build started...
Build succeeded.
No migrations were applied. The database is already up to date.
Done.
PM>
```

A. 13: Setup-6

8. Now run the application from Visual Studio.
9. You will now get the login screen of the system. Please use the below default credentials to log in. You can change the credentials later from the system.
 - Username: Admin
 - Password: 1

APPENDIX B – USER DOCUMENTATION

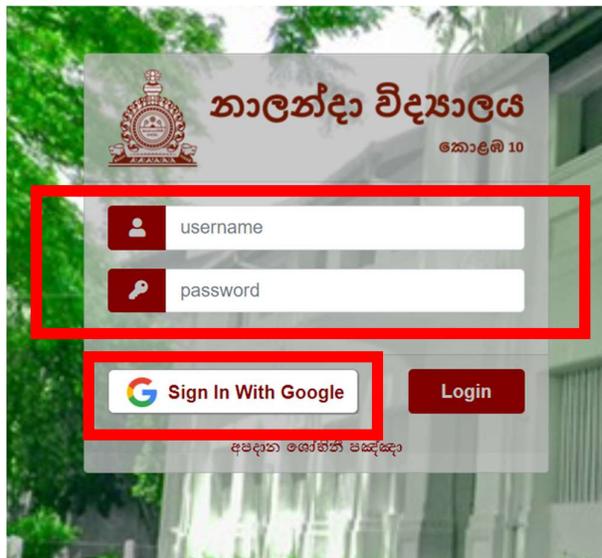
This documentation is to help system users of the school. All kinds of users can use this document for the first time using the system and in any process reference.

User Guide for Admin

Log in

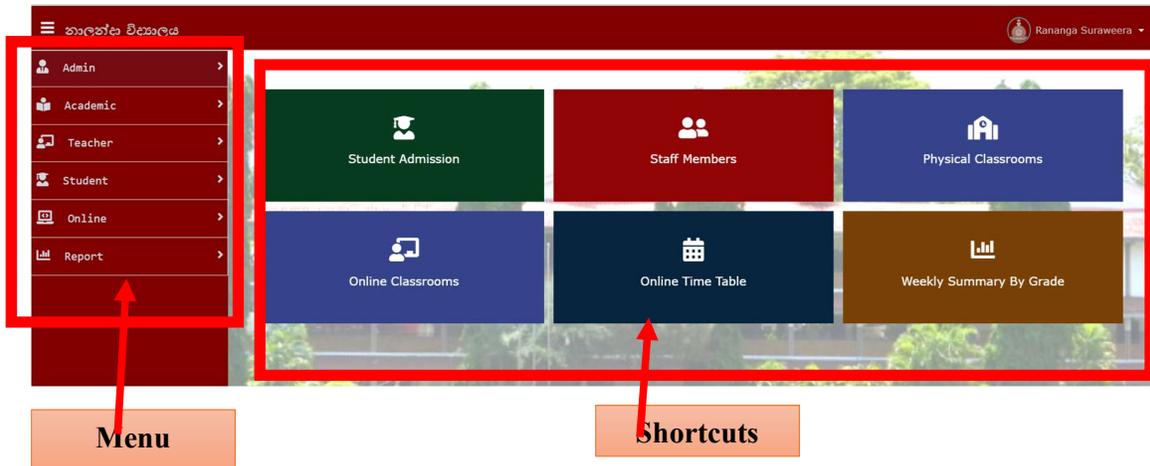
Step 1: Use the provided username and the password in the first login. Once you logged in you can change the password. (User can either use google account to log in)

Figure B1 refers to the steps of login of the system.



B 1: User Login

Step 2: Once successfully logged in you will navigate to the screen shown in Figure B2.



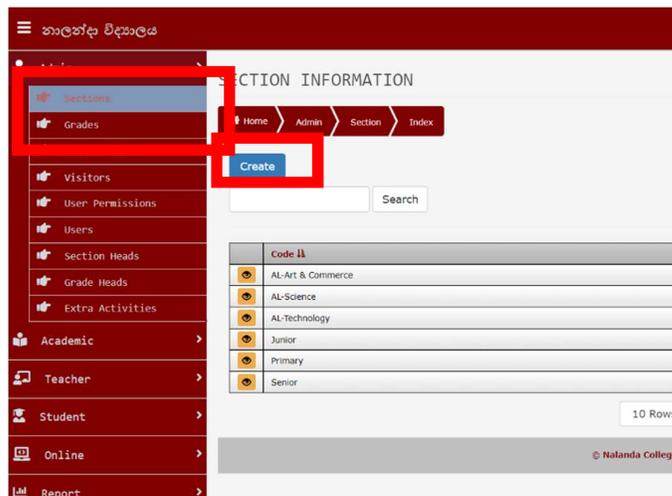
Create Section:

Step 1: Select Admin->Sections from the menu. (Refer Figure B3)

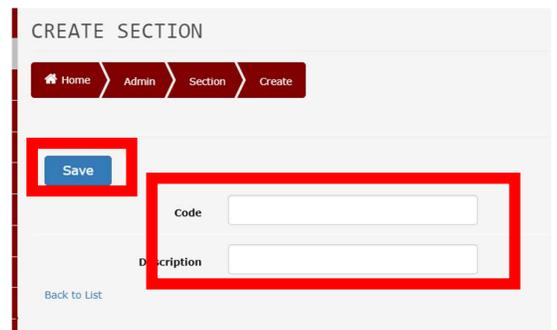
Step 2: Click the Create button to create a new section.

Step 3: Fill in details and save. (Refer Figure B2)

*Created section details can be edited using the  icon.



B 3: Section on menu



B 2: Save Sections

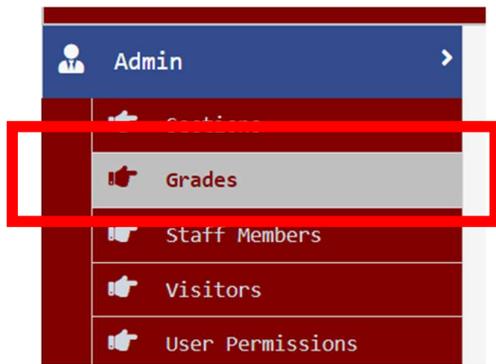
Create Grade:

Step 1: Select Admin->Grades from the menu. (Refer Figure B5)

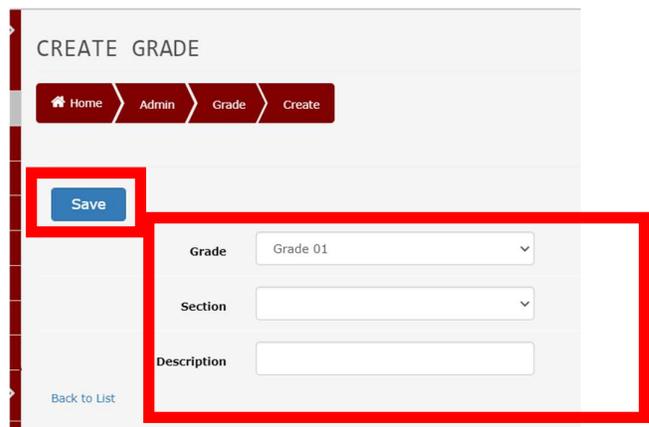
Step 2: Click the Create button to create a new section.

Step 3: Fill in details and save. (Refer Figure B4)

*Created Grade details can be edited using the  icon.



B 5: Grades on Menu



B 4: Save Grades

Create Staff Members:

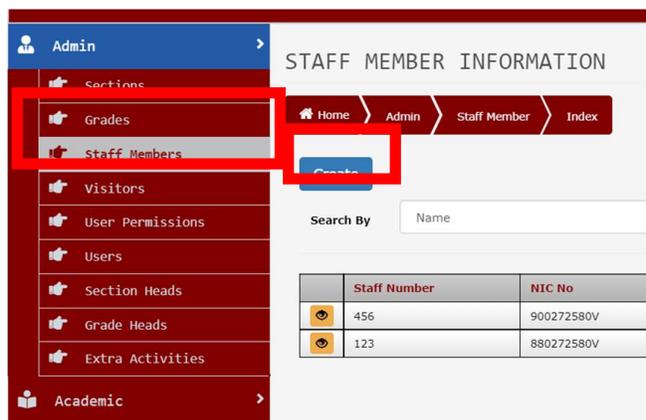
Step 1: Select Admin->Staff Members from the menu. (Refer Figure B6)

Step 2: Click the Create button to create a new section.

Step 3: Fill in details and save. (Refer Figure B7)

*Created Staff Member details can be edited using the  icon.

*All mandatory fields  need to be filled to save the record.



B 6: Staff Members on Menu

CREATE STAFF MEMBER

Home Admin Staff Member Create

Save

Capture a picture using the connected

RETAKE

Staff Number *

Title * Rev

Full Name *

Initials *

Last Name *

Gender * Male

Address1 *

Address2 *

City *

Mobile No *

Home Contact No *

Status Active

Is Teacher

Joined Date *

Emergency Contact Name *

Emergency Contact No *

NIC No *

School Email *

Retired Date *

Back to List

Create Section Heads:

Step 1: Select Admin-> Section Heads from the menu. (Refer Figure B9)

Step 2: Click the Create button to create a new section.

Step 3: Fill in details and save. (Refer Figure B8)

*Created Section Head details can be edited using the  icon.

Admin

Sections

Grades

Staff Members

Visitors

User Permissions

Users

Section Heads

Grade Heads

Extra Activities

SECTION HEAD IN

Home Admin Sect

Create

Search By Year

B 9: Section Heads on Menu

CREATE SECTION HEAD

Home Admin Section Head Create

Save

Year 2021

Section

Section Head --Select--

From Date 2021-01-01

B 8: Save a Section Head

Back to List

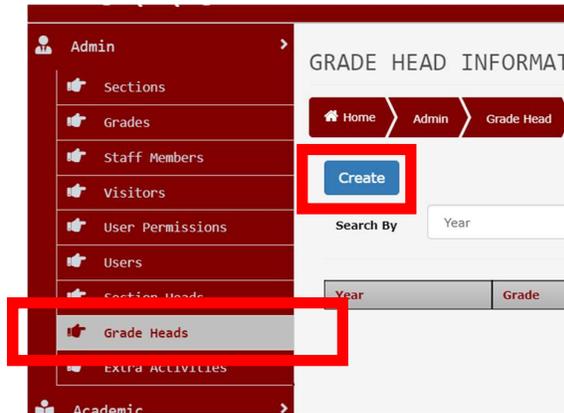
Create Grade Heads:

Step 1: Select Admin-> Grade Heads from the menu. (Refer Figure B11)

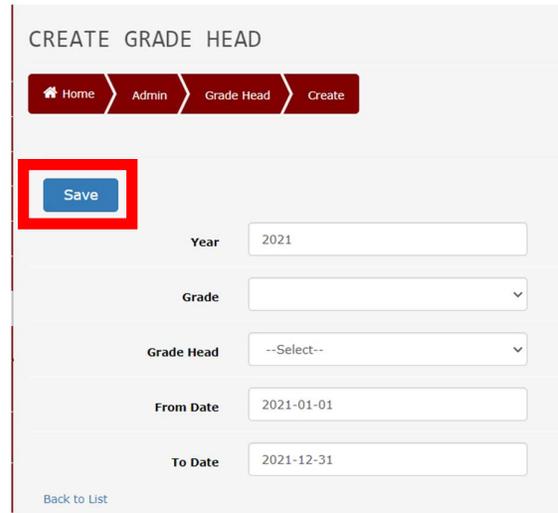
Step 2: Click the Create button to create a new section.

Step 3: Fill in details and save. (Refer Figure B10)

*Created Grade Head details can be edited using the  icon.



B 11: Grade Heads on Menu



B 10: Save a Grade Head Teacher

Create Subject Categories:

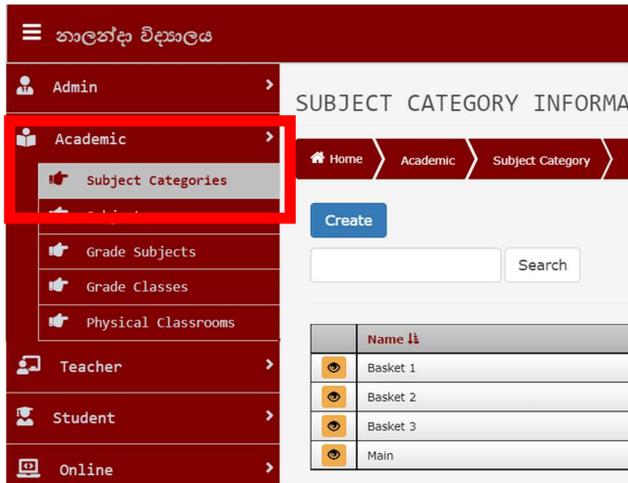
Use the below steps to create categories of subjects. Use this before adding subjects.

Step 1: Select Academic-> Subject Categories from the menu. (Refer Figure B12)

Step 2: Click the Create button to create a new section.

Step 3: Fill in details and save.

*Created category details can be edited using the  icon.



B 12: Subject Categories on menu

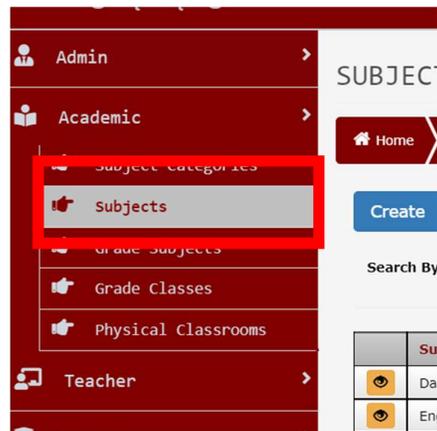
Create Subjects:

Use the below steps to create subjects.

Step 1: Select Academic-> Subjects from the menu. (Refer Figure B13)

Step 2: Click the Create button to create a new section.

Step 3: Fill in details and save.



B 13: Subjects on menu

Create Grade Classes:

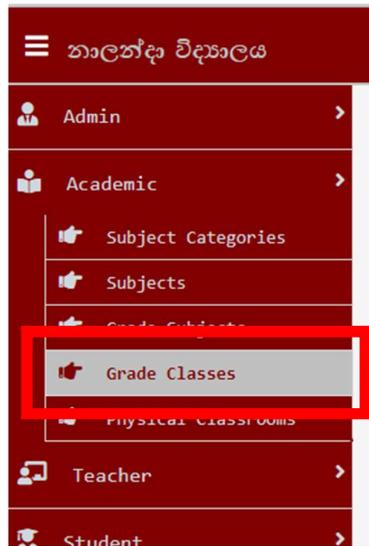
Use the below steps to create grade classes before promoting students.

Step 1: Select Academic-> Grade classes from the menu. (Refer Figure B14)

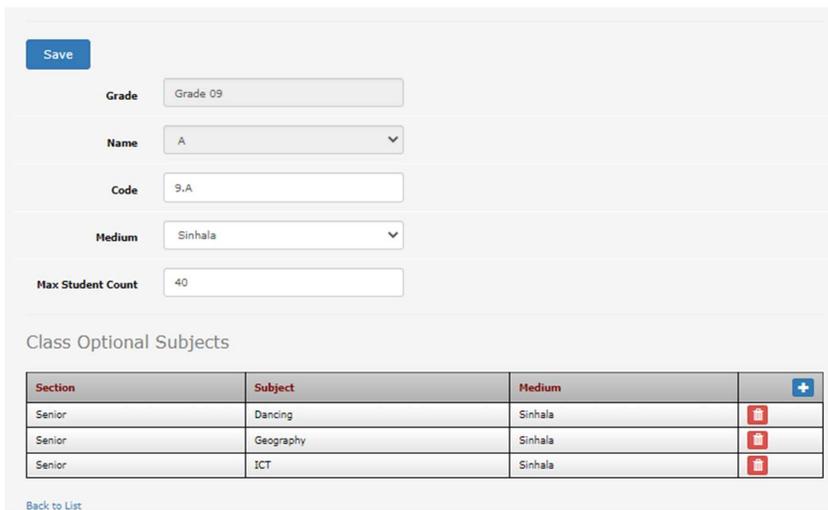
Step 2: Click the Create button to create a new section.

Step 3: Fill in details and save. (Refer Figure B15)

*Class optional subjects can be added using the  button.



B 14: Grade classes on menu

A screenshot of a form for creating a grade class. The form has a 'Save' button at the top left. Below it are several input fields: 'Grade' (Grade 09), 'Name' (A), 'Code' (S.A), 'Medium' (Sinhala), and 'Max Student Count' (40). Below the form is a table titled 'Class Optional Subjects'. The table has columns for 'Section', 'Subject', 'Medium', and a '+' button. The table contains three rows: 'Senior' with 'Dancing', 'Senior' with 'Geography', and 'Senior' with 'ICT'. Each row has a red trash icon in the '+' column. Below the table is a 'Back to List' link.

B 15: Add details and save

Create physical Classes:

Use the below steps to physical classroom details before promoting students.

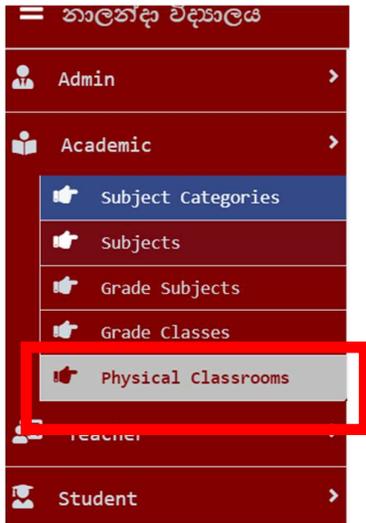
Step 1: Select Academic-> physical classrooms from the menu. (Refer Figure B17)

Step 2: Click the Create button to create a new section.

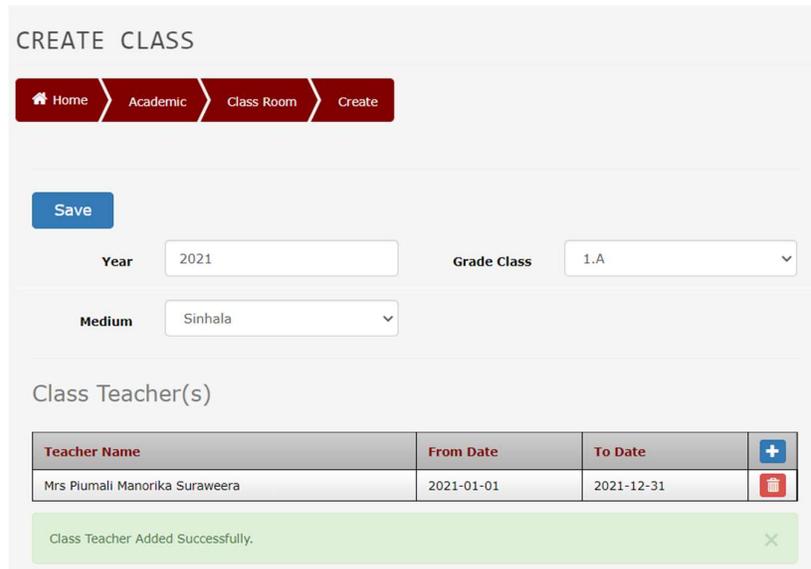
Step 3: Fill in details and save. (Refer Figure B16)

*Class teachers can be added using the  button.

*Added teachers can be deleted using the button  in the list.



B 16: Physical classrooms on the menu



B 17: Create and save the physical classroom details

Add teacher preferred subject details:

Step 1: Select Teacher-> Teacher Subjects from the menu. (Refer Figure B19)

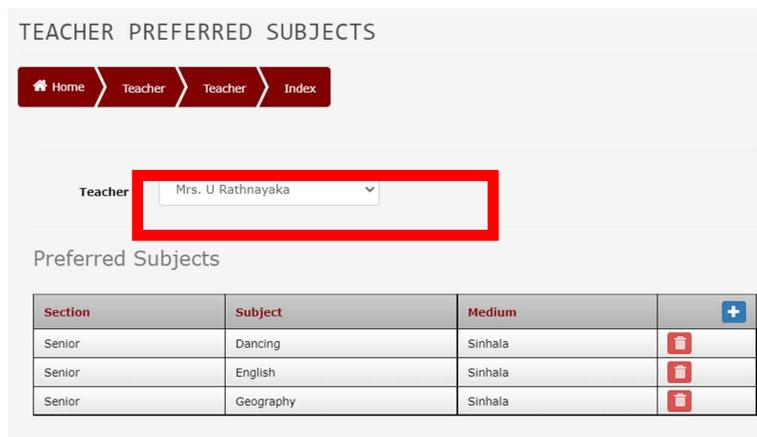
Step 2: Select the teacher from the “Teacher” option.

Step 3: Add preferred subjects to the list using the button . Once the subject is added it will **autosave** to the application. (Refer Figure B18)

*Added subjects can be deleted using the button  in the list.



B 18: Preferred teacher subjects on menu



B 19: Save details

Add qualifications to teachers:

Step 1: Select Teacher-> Teacher Qualifications from the menu. (Refer Figure B21)

Step 2: Select the teacher from the “Teacher” option.

Step 3: Add preferred qualification using the button  through the popup box opened.

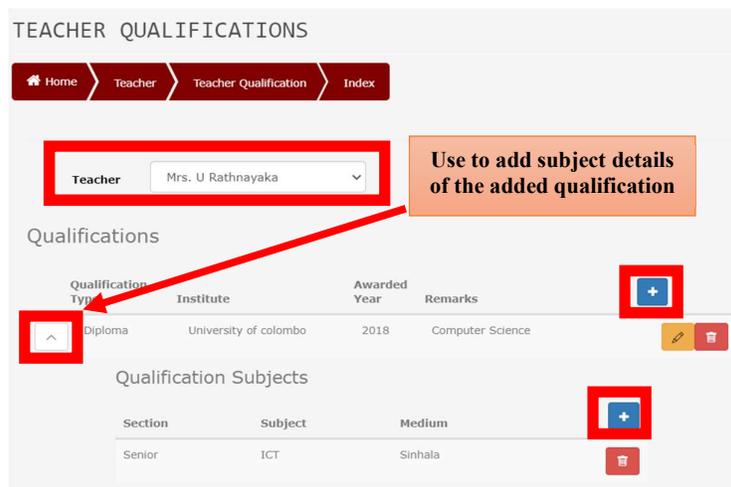
Once the qualification is added it will **autosave** to the application. (Refer Figure B20)

*Added qualifications can be deleted using the button  in the list.

*Added qualifications can be edited using the button  in the list.



B 21:Teacher Qualification on menu



B 20: Add qualifications with subject details

Add off-hours to teachers:

Step 1: Select Teacher-> Teacher off time from the menu. (Refer Figure B23)

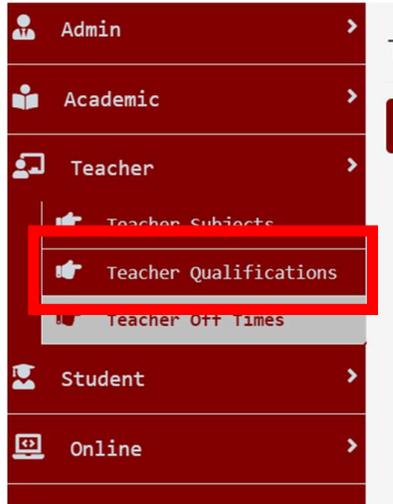
Step 2: Select the teacher from the “Teacher” option.

Step 3: Add a time range with a date using the button  through the popup box opened. (Refer Figure B22)

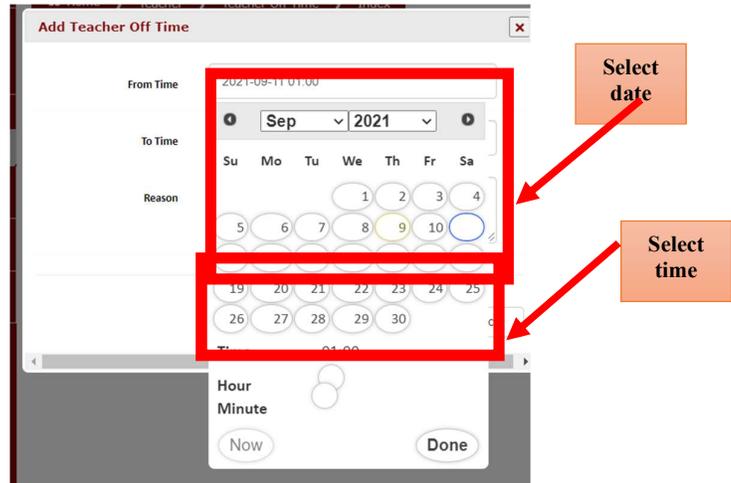
Step 4: Once the date and time are added it will **autosave** to the application. (Refer Figure B24)

*Added times can be deleted using the button  in the list.

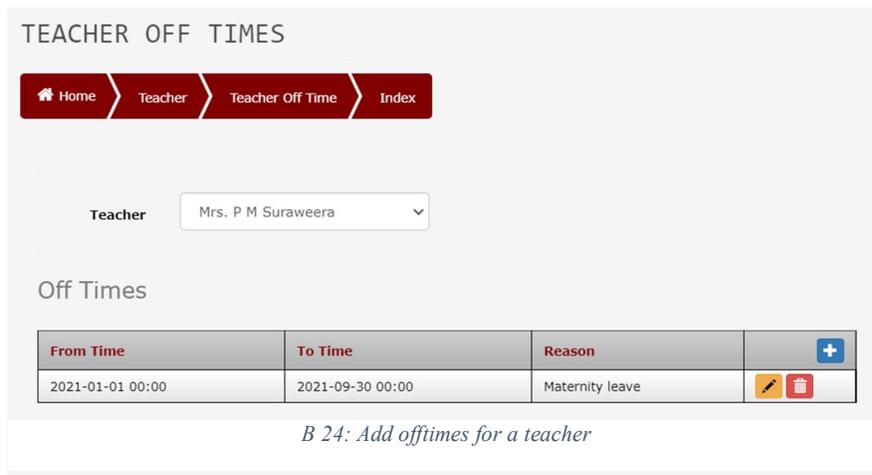
*Added times can be edited using the  button in the list.



B 23: Teacher off-time on menu



B 22: Set date and time on popup



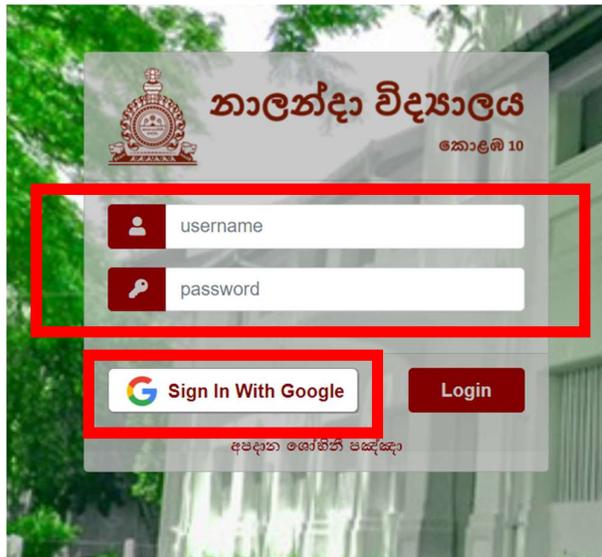
B 24: Add offtimes for a teacher

User Guide for Grade-Head Teachers

Log in

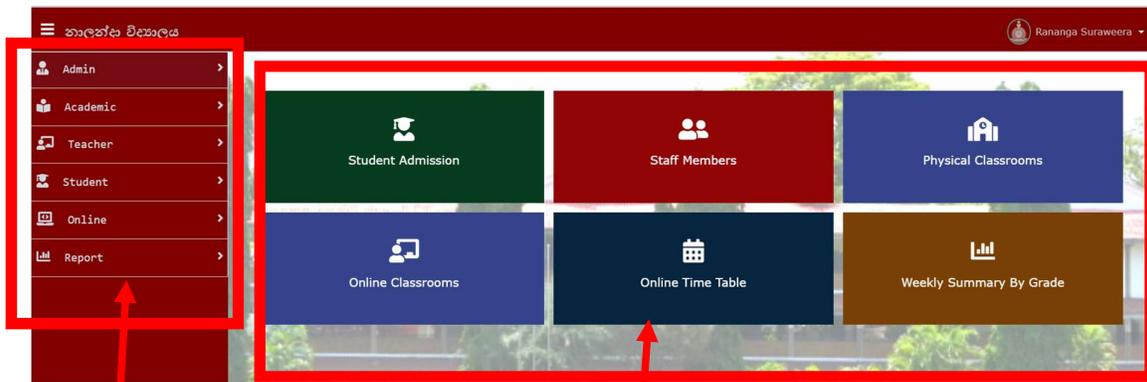
Step 1: Use the provided username and the password in the first login. Once you logged in you can change the password. (User can either use google account to log in)

Figure B25 refers to the steps of login of the system.



B 25: User Login

Step 2: Once successfully logged in you will navigate to the screen shown in Figure B26.



B 26: Main Screen

Menu

Shortcuts

Add a student to the initial class:

Use the below steps to add students to the initial class once after admission.

Do not use this for student promotional purposes.

Step 1: Select Student -> Admit student from the menu. (Refer Figure B27)

Step 2: Select the current year using the “Year” option.

Step 3: Select the admitting grade using the “Grade” option. (Refer Figure B29)

Step 4: Click the  button.

*All pending new admissions will list down and the user will be able to select a class and complete the admission.

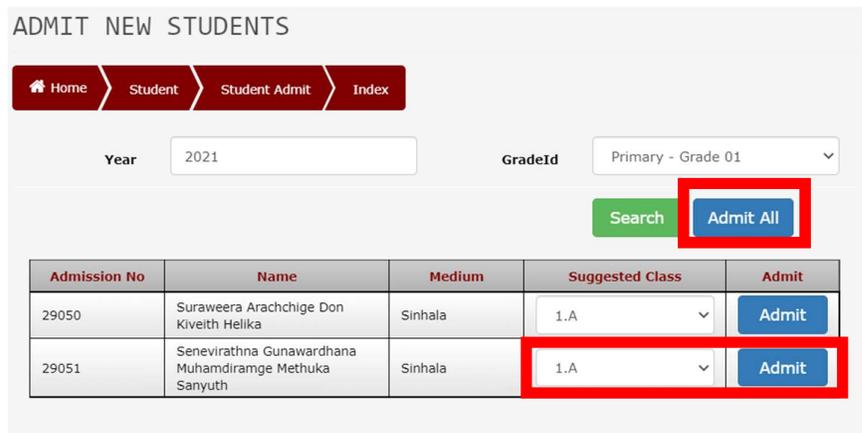
Step 3: Select a class and click the  button. (Refer Figure B28)



B 27: Admit new students on menu



B 29: Select year and grade to admit



B 28: Complete admission

Student class promotion:

Use the below steps to promote students to the next grade level.

Step 1: Select Student -> Class Promotion from the menu. (Refer Figure B30)

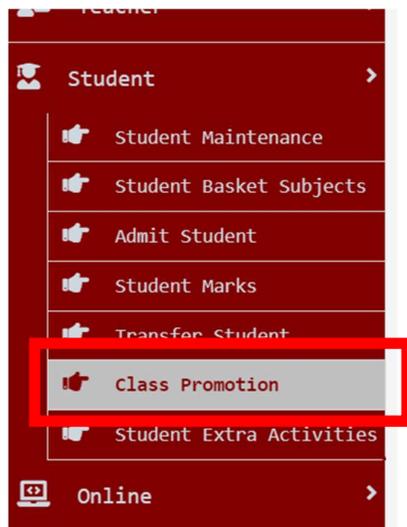
Step 2: Select “Create” for a new promotion.

Step 3: Select promoting to year, from grade, and promoting criteria and click on the “Process” button. (Refer Figure B31)

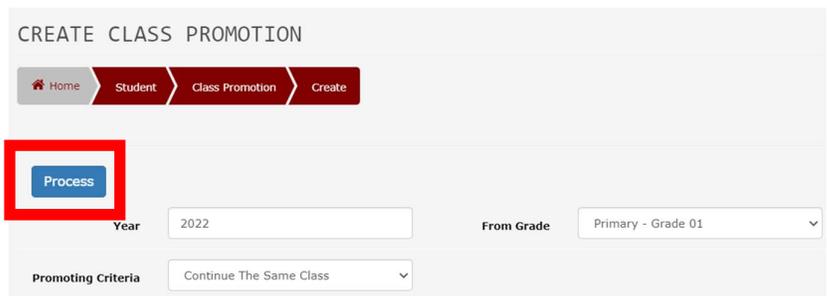
Step 4: Select the promoting class of each student in the list. (Refer Figure B32)

Step 5: Click the “Save button”.

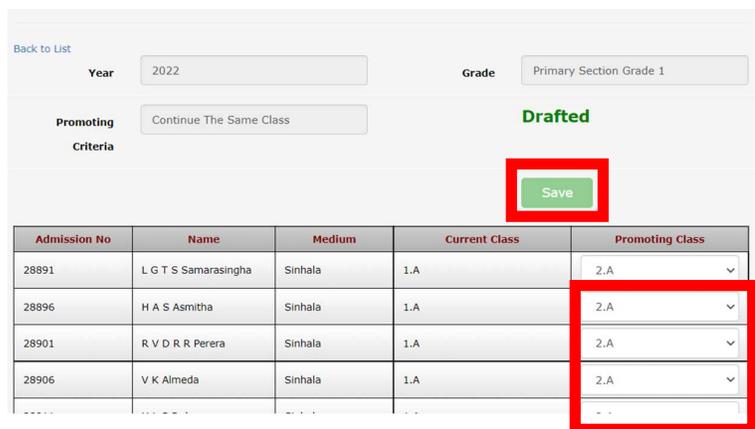
*Created promotion details can be viewed using the  icon.



B 30: Class Promotion on the menu



B 31: Select promoting details



B 32: Completion of promoting

Student transfer:

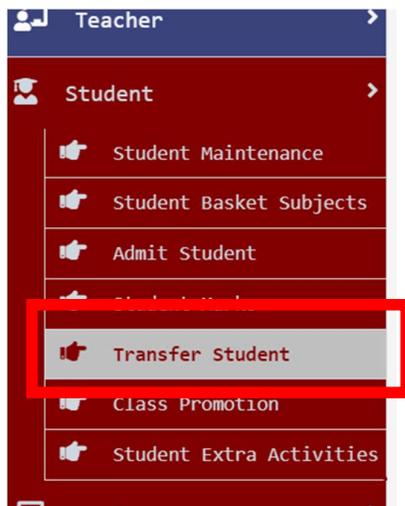
Use the below steps to transfer students to another class.

Step 1: Select Student -> Transfer student from the menu. (Refer Figure B33)

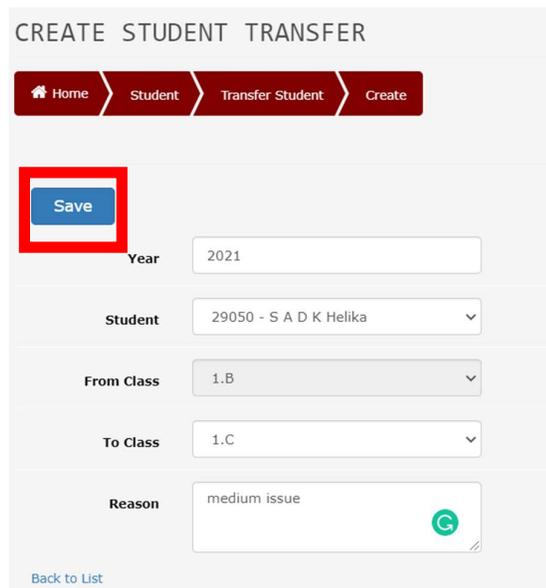
Step 2: Select “Create”.

Step 3: Fill in the data and click on the “Save” button. (Refer Figure B34)

*Created student transfers can be viewed using the  icon.



B 33: Student transfer on menu

A screenshot of a web form titled 'CREATE STUDENT TRANSFER'. The form has a breadcrumb trail: Home > Student > Transfer Student > Create. A blue 'Save' button is highlighted with a red box. The form fields are: 'Year' (text input with value '2021'), 'Student' (dropdown menu with value '29050 - S A D K Helika'), 'From Class' (dropdown menu with value '1.B'), 'To Class' (dropdown menu with value '1.C'), and 'Reason' (text area with value 'medium issue'). There is a green circular icon with a plus sign in the bottom right corner of the reason field. A 'Back to List' link is at the bottom left.

B 34: Complete transfer

Student extra activities:

Use the below steps to add students' extra activities.

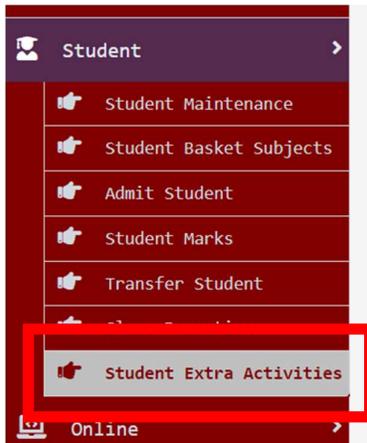
Step 1: Select Student -> Student extra activities from the menu. (Refer Figure B36)

Step 2: Select “Student”.

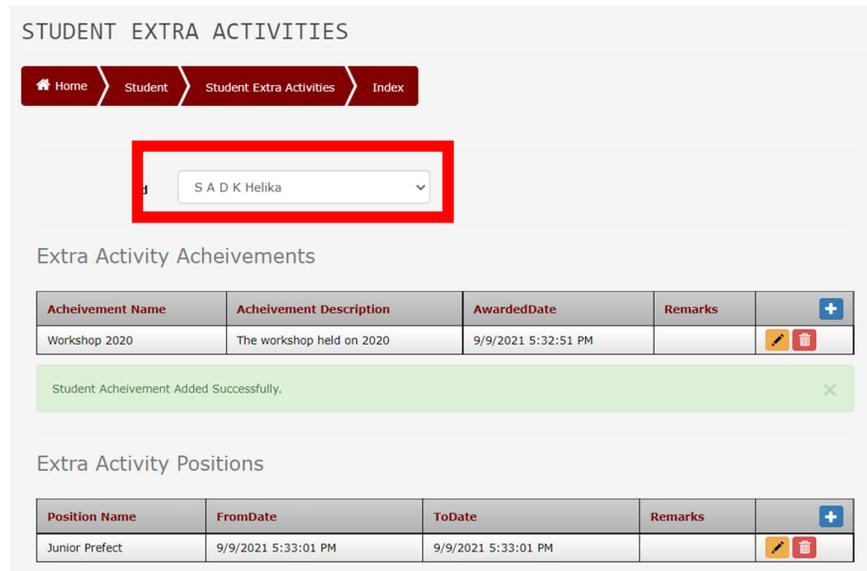
Step 3: Add achievements and positions using the  button. (Refer Figure B35)

*Added achievements and positions can be deleted using the  button in lists.

*Added achievements and positions can be edited using the  button in lists.



B 36: Student extra activities on the menu



B 35: Add achievements and positions

Create online classrooms:

Use the below steps to add online classrooms. Users can add more than one physical class into one online classroom.

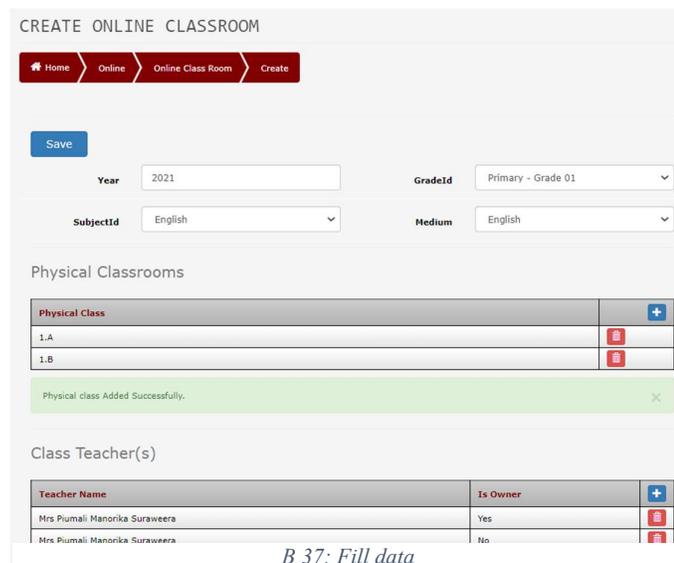
Step 1: Select Online -> Online classrooms from the menu. (Refer Figure B38)

Step 2: Select "Create".

Step 3: Fill in data and "Save".



B 38: Online classrooms on the menu



B 37: Fill data

Create an online timetable:

Use the below steps to add an online timetable for crated online classrooms.

Step 1: Select Online -> Online classrooms from the menu. (Refer Figure B39)

Step 2: Select “Create”.

Step 3: Fill in data and “Save”.



B 39: Online timetable on the menu

APPENDIX C – MIS REPORTS

Management reports are generated via the developed system for the decision-making purpose of the school. Principal, Grade headteachers, and teachers are privileged to generate reports through the system, and reports were designed upon their requests.

Report 1: Online Sessions Summary by Grade

The report in figure C.1 refers to the grade-wise online session summary including student participation. This report is accessible to the Principal and Grade headteachers. The report session data is generated for a given period by user selection.



නාලන්දා විද්‍යාලය, කොළඹ 10

7/2/2021 3:38:42 PM

Online Sessions Summary By Grade

Year : 2021

From Date : 2021-Jun-02

Grade : 6

To Date : 2021-Jul-02

Teacher Name	Class Subject	Total Students	Session Unheld	Session Held	Average Attendance	Average Percentage
Apsara Erandi Vithan	ICT - 8C, 8E	92	0	1	59	64.13 %
B.M.K. Venus	සංගීත - 8B, 8F	88	0	1	32	36.36 %
Chandana Senadeera	English - 8G	48	0	2	39.5	82.29 %
Diana Sladen	English - 8H	49	0	2	29	59.18 %
Dilupa Senevirathna	සංගීත - 8G	48	0	2	40.5	84.38 %
Erangi Munasinghe	පිටු කලාව - 8C, 8D, 8E, 8F	41	0	1	32	78.05 %
Gayathri Srinammuni	ICT - 8B, 8G	90	0	1	68	75.56 %
Geetha Premachandra	සෞඛ්‍ය විද්‍යාව - 8G, 8H	96	0	2	76	79.17 %
Geetha Premachandra	සෞඛ්‍ය විද්‍යාව - 8C, 8D	97	0	2	104	107.22 %
Geetha Premachandra	සෞඛ්‍ය විද්‍යාව - 8E, 8F	97	0	2	84	86.60 %
H.D Nanayakkara	P.T.S - 8A	44	0	1	37	84.09 %
H.D Nanayakkara	මුද්‍රිත චිත්‍රපට - 8D, 8F	98	0	1	67	68.37 %
H.D Nanayakkara	මුද්‍රිත චිත්‍රපට - 8H	50	0	1	30	60.00 %
Harsha Widanage	- Grade 8A	44	0	3	24	54.55 %
Harsha Widanage	English - 8A, 8B	87	0	2	70.5	81.03 %
Indrani Jayasundara	තමිල් - 8A, 8B	86	0	2	67	77.91 %
Indrani Jayasundara	තමිල් - 8C, 8D	93	0	2	61.5	66.13 %
Indrani Jayasundara	තමිල් - 8E, 8F	95	0	2	77.5	81.58 %
Indrani Jayasundara	තමිල් - 8G, 8H	92	0	2	63	68.48 %
Jayantha Warusavithana	- Grade 8G	48	0	3	23.67	49.31 %
Jayantha Warusavithana	සංගීත - 8G	48	0	4	36	75.00 %
Jayantha Warusavithana	සංගීත - 8E, 8F	96	0	3	83.67	87.16 %
K. V. S. Priyadarshani	විද්‍යාව - 8D, 8E	101	0	3	82.33	81.51 %
M K Chandralatha	සංගීත - 8E	52	0	2	45	86.54 %
M K Chandralatha	සුවිද්‍යා සංගීත - 8C, 8H	91	0	1	67	73.63 %
Manel Senarathne	සංගීත සංගීත - Grade 8	99	0	1	77	77.78 %

Report 2: Online Session Attendance

The report in figure C.2 refers to the Online Session Attendance of each class. The report permission is granted to the Principal, Grade head, and Class teachers. This report can be generated for a given period in each class by the user selection.



නාලන්දා විද්‍යාලය, කොළඹ 10

7/2/2021 3:49:39 PM

Online Session Attendance

Class : Grade 1A

Year : 2021

From Date : 2021-Jun-02

Teacher : Subeda Nawarathna

Grade : 1

To Date : 2021-Jul-02

Student	2021-06-01 Grade 1	2021-06-03 Grade 1	2021-06-04 Grade 1	2021-06-07 Grade 1	2021-06-08 Grade 1	2021-06-10 Grade 1
A J M T A Jayasundara	✓	✗	✗	?	?	?
A U N De Silva	✗	✗	✗	?	?	?
B L R Abedeera	✗	✗	✗	?	?	?
B M T Silva	✗	✗	✗	?	?	?
B W K M Peeris	✗	✗	✗	?	?	?
Chenuk Manthila P S	✗	✗	✗	?	?	?
D C A Dias	✗	✗	✗	?	?	?
D J M K N Serasingha	✗	✗	✗	?	?	?
D M K S Liyanage	✗	✗	✗	?	?	?
D M L Dasanayake	✗	✗	✗	?	?	?
D N Gunasena	✗	✓	✗	?	?	?
D W O D De Silva	✗	✗	✗	?	?	?
E H H Nethsara	✗	✗	✗	?	?	?
H A S Asmitha	✗	✗	✗	?	?	?
H L S Dulsara	✗	✗	✗	?	?	?
I U Roopasingha	✗	✗	✗	?	?	?
K A D Sanketh	✗	✗	✗	?	?	?
K A M Menath	✗	✗	✗	?	?	?
L G T S Samarasingha	✗	✗	✗	?	?	?
M K G Darmasiri	✗	✗	✗	?	?	?
N H E De Silava	✗	✗	✓	?	?	?
R V D R R Perera	✗	✗	✗	?	?	?
S A J Pathirana	✗	✗	✗	?	?	?
S H V Sanith	✗	✗	✗	?	?	?
S I Kiriwandeniya	✗	✗	✗	?	?	?
S L D Karunathilaka	✗	✗	✗	?	?	?
S O Leelarathna	✗	✗	✗	?	?	?

1

C. 2: Online Session Attendance Report by Class

Report 3: Online Teaching Weekly Summary

The report in figure C.3 refers to the weekly summary of online sessions of each grade. This report is accessible to the Principal and Grade headteachers. the report can be generated for a given period of given grade by a user selection.



නාලන්දා විද්‍යාලය, කොළඹ 10
Online Teaching Weekly Summary

7/2/2021 3:03:56 PM

Year : 2021
Grade : 1

From Date : 2021-Jun-01
To Date : 2021-Jun-30

දිනය	වේලාව	විෂය	පාඨය/පාඨි	ඉගැන්වූ පාඨම	සිසුන් ගණන	ගුරුභාවතාගේ නම
2021-06-01	ප.ව. 05:00 - ප.ව. 06:30	Grade 1	1.A	Grade 1A - Session 12	1 / 32	Piumali Manorika Suraweera
2021-06-03	ප.ව. 05:00 - ප.ව. 06:30	Grade 1	1.A	Grade 1A - Session 13	1 / 32	Piumali Manorika Suraweera
2021-06-04	ප.ව. 05:00 - ප.ව. 06:30	Grade 1	1.A	Grade 1A - Session 14	1 / 32	Piumali Manorika Suraweera
2021-06-07	ප.ව. 05:00 - ප.ව. 06:30	Grade 1	1.A	බුද්ධ ධර්මය, පුවර් භාෂා කුසලතා, ගණිත සංකල්ප, පරිසරය	0 / 32	Piumali Manorika Suraweera
2021-06-08	ප.ව. 05:00 - ප.ව. 06:30	Grade 1	1.A	පරිසරය, පුවර් භාෂා කුසලතා, ගණිත සංකල්ප, පෙර ලිඛිත	0 / 32	Piumali Manorika Suraweera
2021-06-10	ප.ව. 05:00 - ප.ව. 06:30	Grade 1	1.A	බුද්ධ ධර්මය, පුවර් භාෂා කුසලතා, ගණිත සංකල්ප, පෙර ලිඛිත, පරිසරය	0 / 32	Piumali Manorika Suraweera
2021-06-11	ප.ව. 05:00 - ප.ව. 06:30	Grade 1	1.A	පුවර් භාෂා කුසලතා, ගණිත සංකල්ප, පෙර ලිඛිත, පරිසරය	0 / 32	Piumali Manorika Suraweera
2021-06-14	ප.ව. 05:00 - ප.ව. 06:30	Grade 1	1.A	බුද්ධ ධර්මය, මර්ධස, ගණිත සංකල්ප, පරිසරය	0 / 32	Piumali Manorika Suraweera
2021-06-15	ප.ව. 05:00 - ප.ව. 06:30	Grade 1	1.A	පරිසරය, මර්ධස, ගණිත සංකල්ප, බුද්ධ ධර්මය	0 / 32	Piumali Manorika Suraweera
2021-06-17	ප.ව. 05:00 - ප.ව. 06:30	Grade 1	1.A	බුද්ධ ධර්මය, මර්ධස, ගණිත සංකල්ප, පරිසරය	0 / 32	Piumali Manorika Suraweera
2021-06-18	ප.ව. 05:00 - ප.ව. 06:30	Grade 1	1.A	මර්ධස, ගණිත සංකල්ප, බුද්ධ ධර්මය, පරිසරය	0 / 32	Piumali Manorika Suraweera

C. 3: Online weekly summary report

Report 4: Students Term Wise Total Marks

The report in figure C.4 refers to the Grade-wise student marks report. The Grade headteachers are permitted to this report access. The report can be generated by filtering the grade for a particular year.



නාලන්දා විද්‍යාලය, කොළඹ 10

7/2/2021 3:58:04 PM

Students Term Wise Total Marks

Year : 2021

Order By : Term 2 Marks

Grade : 6

Admission No	Student Name	Current Class	Term 1 Marks	Term 2 Marks	Term 3 Marks	Average Marks
27626	D. K. Yenula Manumitha Dissanayake	6B	895	895	895	895
28025	E. E. A. P. Karunaratne	6C	891	891	891	891
27611	P. D. B. Devmith Padukka	6C	889	889	889	889
27720	K. A. V. Hemsandu Thilakarathna	6B	886	886	886	886
27641	K. P. A. Vinuka Kathriarachchi	6B	884	884	884	884
27696	Saman Dissanayake	6C	878	878	878	878
27696	P. D. B. Devmith Padukka	6B	877	877	877	877
28414	Vihara Dintharu Karunaratne	6B	873	873	873	873
27630	T. V. D. Akindu Mandiw Vitharana	6A	868	868	868	868
27601	M. Aritha Gunawardana	6C	867	867	867	867
28882	Ositha H. Gunasekara J. H.	6B	862	862	862	862
27730	H. K. S. Dinsara Habaraduwa	6A	859	859	859	859
27657	S. A. D. R. Vilan Samarantunge	6C	856	856	856	856
27616	H.S.C.Ayan Perera	6A	851	851	851	851
27746	C. Devmeth Jayawardana	6B	851	851	851	851
27706	D. S. D. Sachintha Sekara	6C	846	846	846	846
27602	M. Sithula Gunawardana	6C	845	845	845	845
28533	S. D. Epa Seneviratne	6A	842	842	842	842
27751	T. M. Thisuka D. Rodrigo	6A	840	840	840	840
27715	H. Danuja D. Hettiarachchi	6B	840	840	840	840
27676	U. P. A. Keshawa D. Amarasinghe	6A	839	839	839	839
27696	B. A. T. Sendika Balasuriya	6C	835	835	835	835
27666	K. M. D. Boseeth Kaluwila	6C	834	834	834	834
27691	V. Linash Alahendra	6A	833	833	833	833

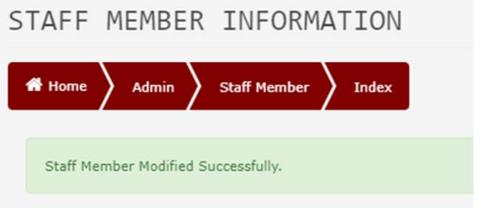
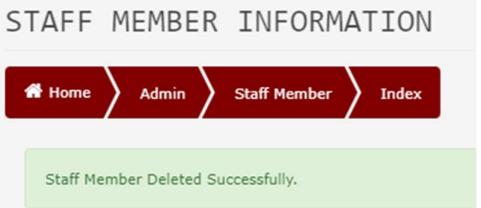
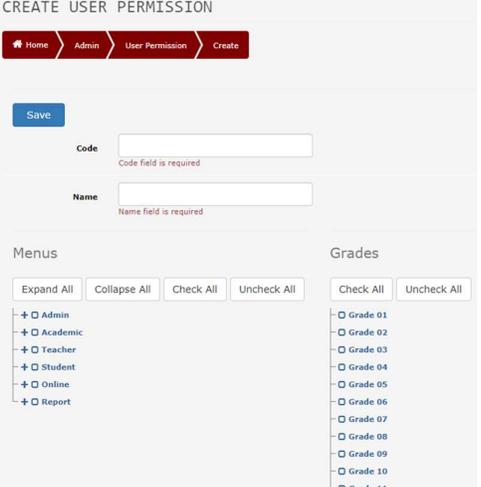
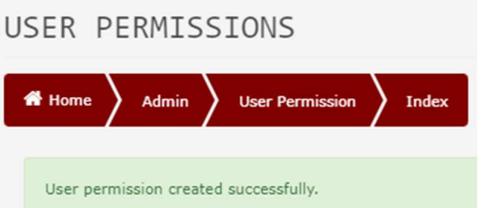
1

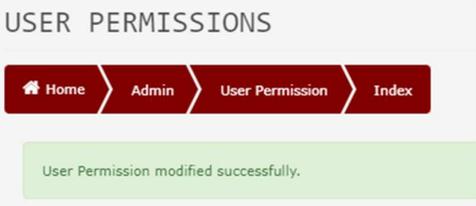
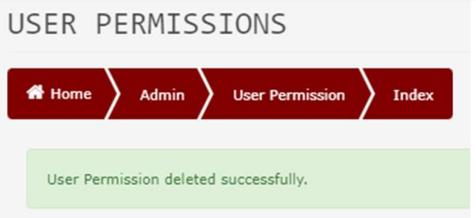
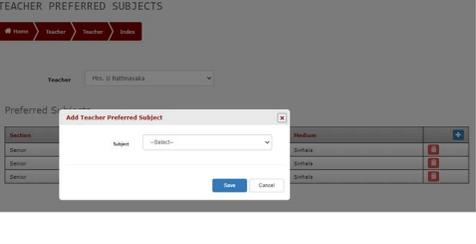
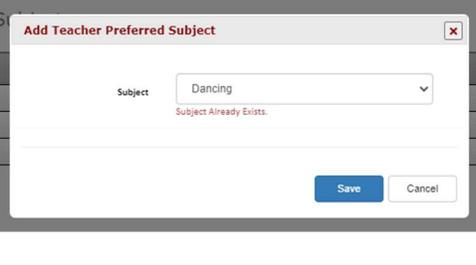
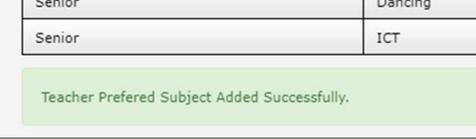
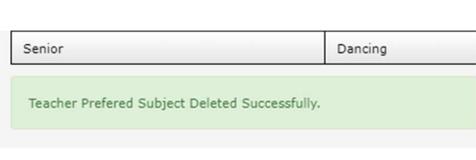
C. 4: Student Marks Report

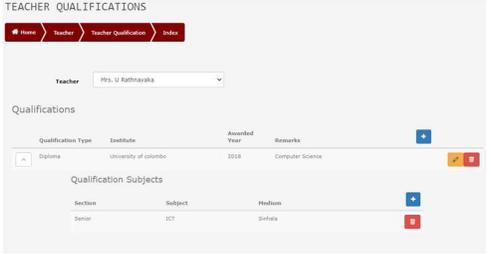
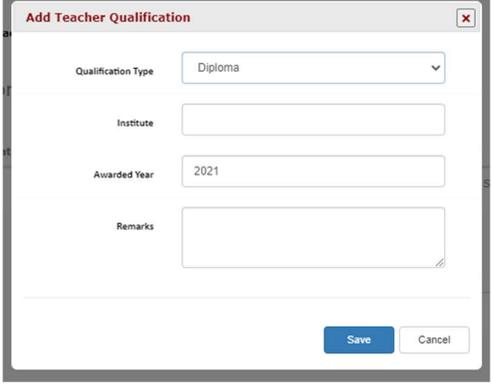
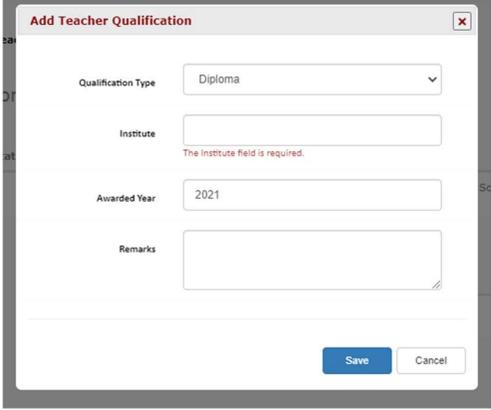
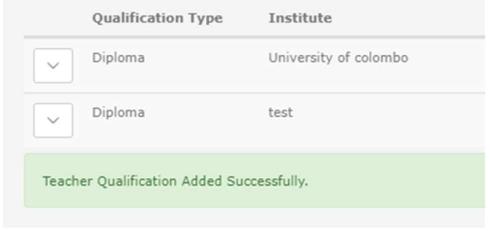
APPENDIX D – TEST RESULTS

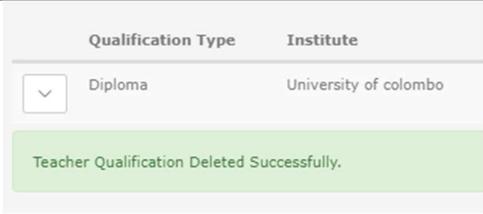
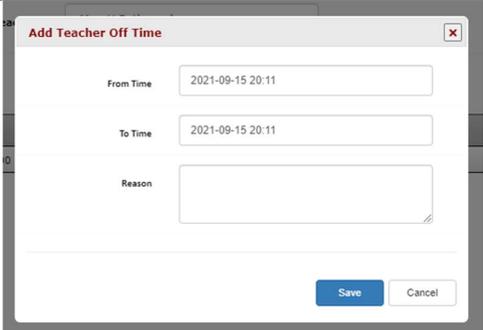
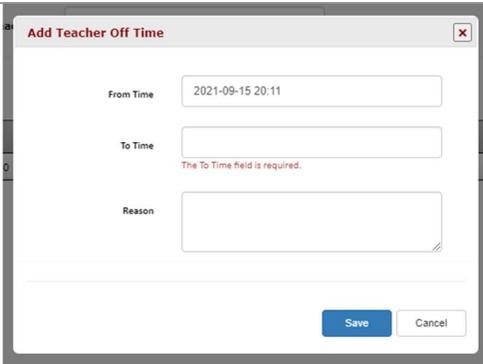
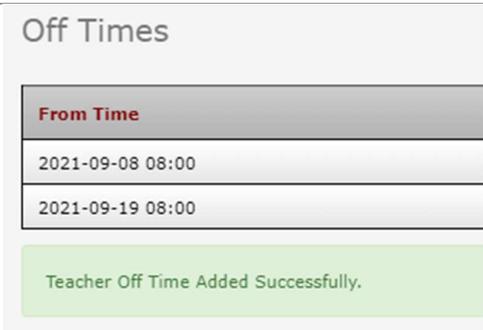
Table D.1 refers to the test results with evidence for additional test cases of the developed system.

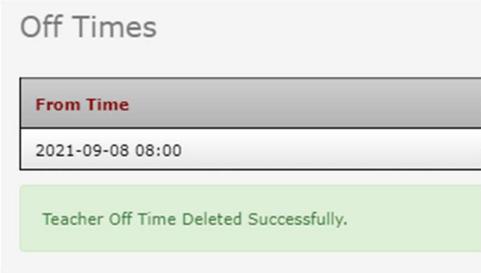
Test #	Test Description	Input Values	Expected Outcome	Result
Staff Members				
1	Click on the staff members menu	None	The system should list all the defined staff members and should be able to filter.	
2	Visit the create page and create a new staff member	Mandatory fields are not filled out	Error messages should appear under all the required fields	
3	Visit the create page and create a new staff member	Enter an existing staff number	Error messages should appear saying staff number already exists.	
4	Visit the create page and create a new staff member	Capture new image	The camera is switched on and should be able to capture an image.	

5	Visit the create page and create a new staff member	Fill out all the details	Should be able to save without an issue.	
6	Select a staff member, visit the edit page, and modify some info	Change address	Should be able to save without an issue.	
7	Select a staff member and delete it	None	The record should be deleted	
User Permissions				
8	Click on the user permissions menu	None	The system should list all the defined user permissions and should be able to filter.	
9	Visit the create page and create a new user permission	Mandatory fields are not filled out	Error messages should appear under all the required fields	
10	Visit the create page and create a new user permission	Fill out all the details	Should be able to save without an issue.	

11	Select user permission, visit the edit page and modify some info	Uncheck some menus and grades.	Should be able to save without an issue.	
12	Select user permission and delete it	None	The record should be deleted	
Teacher Preferred Subjects				
13	Click on the teacher subjects menu	Select a teacher	The system should list all the defined preferred subjects for a teacher.	
14	Select a teacher, click on the add button	None	A popup dialogue should be prompted.	
15	Select a teacher, click on the add button	Select an existing subject	Error messages should appear saying the subject already exists.	
16	Select a teacher, click on the add button	Fill out with valid details	Should be able to save without an issue.	
17	Select a teacher, click on the delete button for a subject	None	Delete confirmation message should appear.	
18	Select a teacher, click on the delete button for a subject and confirm.	None	The record should be deleted	
Teacher Qualifications				

19	Click on the teacher qualifications menu	Select a teacher	The system should list all the defined qualifications for a teacher.	 <p>TEACHER QUALIFICATIONS</p> <p>Home > Teacher > Teacher Qualification > Index</p> <p>Teacher: Mrs. U Rathnayaka</p> <p>Qualifications</p> <table border="1"> <thead> <tr> <th>Qualification Type</th> <th>Institute</th> <th>Awarded Year</th> <th>Remarks</th> </tr> </thead> <tbody> <tr> <td>Diploma</td> <td>University of colombo</td> <td>2018</td> <td>Computer Science</td> </tr> </tbody> </table> <p>Qualification Subjects</p> <table border="1"> <thead> <tr> <th>Section</th> <th>Subject</th> <th>Medium</th> </tr> </thead> <tbody> <tr> <td>Senior</td> <td>ICT</td> <td>Sinhala</td> </tr> </tbody> </table>	Qualification Type	Institute	Awarded Year	Remarks	Diploma	University of colombo	2018	Computer Science	Section	Subject	Medium	Senior	ICT	Sinhala
Qualification Type	Institute	Awarded Year	Remarks															
Diploma	University of colombo	2018	Computer Science															
Section	Subject	Medium																
Senior	ICT	Sinhala																
14	Select a teacher, click on the add button	None	A popup dialogue should be prompted.	 <p>Add Teacher Qualification</p> <p>Qualification Type: Diploma</p> <p>Institute: <input type="text"/></p> <p>Awarded Year: 2021</p> <p>Remarks: <input type="text"/></p> <p>Save Cancel</p>														
15	Select a teacher, click on the add button	Mandatory fields are not filled out	Error messages should appear under all the required fields	 <p>Add Teacher Qualification</p> <p>Qualification Type: Diploma</p> <p>Institute: <input type="text"/> <small>The Institute field is required.</small></p> <p>Awarded Year: 2021</p> <p>Remarks: <input type="text"/></p> <p>Save Cancel</p>														
16	Select a teacher, click on the add button	Fill out with valid details	Should be able to save without an issue.	 <p>Qualification Type: Diploma</p> <p>Institute: University of colombo</p> <p>Qualification Type: Diploma</p> <p>Institute: test</p> <p>Teacher Qualification Added Successfully.</p>														
17	Select a teacher, click on the delete button for a qualification	None	Delete confirmation message should appear.	 <p>Confirm teacher qualification deletion</p> <p>Are you sure you want to delete?</p> <p>Delete Cancel</p>														

18	Select a teacher, click on the delete button for qualification and confirm.	None	The record should be deleted	
Teacher Off Times				
19	Click on the teacher off times menu	Select a teacher	The system should list all the defined off times for a teacher.	
20	Select a teacher, click on the add button	None	A popup dialogue should be prompted.	
21	Select a teacher, click on the add button	Mandatory fields are not filled out	Error messages should appear under all the required fields.	
22	Select a teacher, click on the add button	Fill out with valid details	Should be able to save without an issue.	

23	Select a teacher, click on the delete button for an off-time	None	Delete confirmation message should appear.	
24	Select a teacher, click on the delete button for off-time and confirm.	None	The record should be deleted	

D 1: Additional Test Results