

NUST BOOK



By

NC Hassan Bin Faiz

PC Waqar Ahmed

PC Zubair Majeed

PC Abdul Moueed

Supervisor:

Asst. Prof. Athar Mohsin Zaidi

Submitted to the Faculty of Computer Science

National University of Science and Technology, Rawalpindi

For the partial requirement of Bachelors of Engineering in Computer Software Engineering

July 2015

CERTIFICATE

Certified that the contents and the form of project report entitled “**NUST BOOK**” submitted by Abdul Moueed, Zubair Majeed, Waqar Ahmed and Hassan Faiz has been found satisfactory for the requirement of the degree.

(Supervisor Name)

_____ Department

MCS

Dated

13th May 2015

ABSTRACT

The students of NUST face many problems when accessing the current LMS- the moodle system. It's not responsive for use on cell phone devices. It doesn't allow uploading of assignments and notes via cell phone devices.

NUST BOOK is a web application that provides learning management to the students. It shall also manage the day to day activities, student's progress. Although every university has a functional learn management portal but that is too generic and can't be focused on each student simultaneously. NUST BOOK is specifically developed for students and puts focus on every student's personalized academic needs.

NUST BOOK is a web application which focuses on customized user needs, provides specific details, evaluations and updates to students like their courses attendances, Short attendances, Grades, Upcoming academic activities or subjects they are supposed to take. It also provides personalized information like policies and university regulations to the students that are applicable to them. As a bright student would not need information on repeating a course or clearing an F grade, a struggling student would not want to know how to get a gold medal in academics or final year projects. NUST BOOK is designed to provide the information related to the student using the application. This way it helps student keep track of all the things related to them and save their time by giving them the related information only.

DECLARATION

No portion of the work presented in this dissertation has been submitted in support of another award or qualification either at this institution or elsewhere.

DEDICATION

To Our Parents and Teachers for their continuous support

ACKNOWLEDGMENTS

We are grateful to our parents for their support and prayers. Their faith kept us going.

We are extremely grateful to our project supervisor Asst Prof Athar Mohsin Zaidi for his support and guidance without which we could not have completed the project.

We are thankful to our teachers and instructors here at Computer Science Department, MCS, all of them have guided us to make it possible for us to complete both this degree and the project.

Table of Contents

Contents

1. Introduction.....	1
1.1 Introduction.....	1
1.2 Background.....	1
1.3 Problem Addressed.....	1
1.4 Goals and Objectives.....	2
1.5 Deliverables.....	2
1.6 Organization of the Document.....	2
2. Literary Review.....	3
2.1 Introduction.....	3
2.2 Previous Work.....	3
2.3 Shortcomings.....	4
2.4 Conclusion.....	4
3. System Requirements.....	5
3.1 Introduction.....	5
3.2 Purpose.....	5
3.3 Intended Audience and Reading Suggestions.....	5
3.4 Product Scope.....	5
3.5 Overall Description.....	6
3.5.3 User Classes and Characteristics.....	8
3.5.4 Operating Environment.....	9
3.5.5 Design and Implementation Constraints.....	9
3.5.6 User Documentation.....	9
3.5.7 Assumptions and Dependencies.....	9
3.6 External Interface Requirements.....	10
3.6.1 User Interfaces.....	10
3.6.2 Hardware Interfaces.....	11
3.6.3 Software Interfaces.....	11
3.6.4 Communications Interface.....	12
3.7 System Features.....	12
3.7.1 Student Progress.....	12
3.7.2 LMS.....	13
3.7.3 Record Book.....	13
3.7.4 Attendance.....	13
3.7.5 Upcoming Events.....	14
3.7.6 NUST Policies.....	14
3.8 Nonfunctional Requirements.....	14
3.8.1 Performance Requirements.....	14
3.8.2 Safety Requirements.....	15
3.8.3 Security Requirements.....	15
3.8.4 Software Quality Attributes.....	15
3.9 Conclusion.....	15
4. System Design Specifications.....	16
4.1 Introduction.....	16
4.2 Purpose.....	16
4.3 Product Scope.....	16
4.4 Intended Audience and Reading Suggestions.....	16
4.5 Document Overview.....	17
4.6 System Architecture Description.....	17
4.7.1 Overview of modules.....	17
4.7.2 Structure and relationship.....	33
4.8 Conclusion.....	59
5. Project Analysis and Evaluation.....	60
5.1 Introduction.....	60
5.2 Testing.....	60

5.3	Testing Levels	60
5.4	Unit Testing	60
6.	Conclusion and Future Work	71
	Appendix A: User Manual	72
7.	General Information	1
7.1	System Overview	1
7.2	Project References	1
7.3	Organization of the Manual	1
7.4	Acronyms and abbreviations	2
8.	System Summary	3
8.1	System Configuration	3
8.2	User Access Levels	3
8.3	Contingencies and Alternate mode of Operations	3
9.	Using the System	4
9.1	Logging in	4
9.2	Successful Login	4
	Appendix B: References	5

List of Figures

Figure Name	Pg. No.
Figure 3.1: Context Diagram	6
Figure 3.2: Use Case Diagram	8
Figure 4.1: MVC	18
Figure 4.2: Class Diagram	33
Figure 4.3: Login Sequence	34
Figure 4.4: Attendance Sequence	35
Figure 4.5: Grades Sequence	36
Figure 4.6: Subject Registration Sequence	37
Figure 4.7: Improvements Sequence	38
Figure 4.8: Balanced Sequence	39
Figure 4.9: Internship/Scholarship Sequence	40
Figure 4.10: Events Sequence	41
Figure 4.11: Activity Diagram	42
Figure 4.12: State Machine Diagram	43
Figure 4.13: Main Use Case Diagram	44

List of tables

Table Name	Pg. No.
Table 4.1: Component Template Description	28
Table 4.2: Database.....	29
Table 4.3: Login.....	31
Table 4.4: Student	32
Table 4.5: User.....	33
Table 4.6: Faculty	34
Table 4.7: Subject	35
Table 4.8: Message	36
Table 4.9: Post	37
Table 4.10: Group	38
Table 4.11: Events	39
Table 4.12: Login.....	40
Table 4.13: Attendance Check.....	41
Table 4.14: Upload Check	42
Table 4.15: Subject Registration.....	42
Table 4.16: Grades check.....	43

1. Introduction

1.1 Introduction

This chapter gives an overview and Introduction of the NUST BOOK LMS system. It gives a background and reason to develop the system as well as the solution we have managed to develop to the respective problem.

1.2 Background

Previously implemented LMS system is used by the students and faculty of NUST. However it is sometimes difficult to use and does not provide additional features that might help students.

Previously NUST-LMS (Learning Management System) is working and all the NUST (National University of Sciences and Technology) colleges are using it. It is a web application. NUST-LMS is working and all the NUST colleges are using it.

1.3 Problem Addressed

There is need to develop a learning management system that has to offer much more.

There should be more features included that will aid the students and will provide a better environment including easy and understandable interface and features like chat and social interaction of students and faculty through LMS.

NUST Book is an application that will aid in accessing the LMS. This application shall guide the student's through-out their four-year degree program. NUST Book is an application that will aid in accessing the LMS. This application shall guide the student's through-out their four-year degree program.

The application shall keep track of the students' progress and attendance of the current semester as well as the previous semesters. The application shall provide guidelines to the students on how to improve their results. The application will use web scraping, html parsing to create an intuitive interface for exploring the LMS. The application will access LMS to

fetch data required for the application to operate properly and therefore the application requires the use of internet to be fully functional.

1.4 Goals and Objectives

Our goal is to develop a Learning management system that is easy and has a friendly interface and provides a lot more additional features that are not included in the previously implemented LMS system.

1.5 Deliverables

The deliverables of the project include Project Synopsis, Software Requirement Specification, Software Design Specification and Web Application

1.6 Organization of the Document

The document is divided into 7 chapters. The first chapter introduces the project and is an introduction to the document itself. The second chapter provides an overview and detail of the literary review. The Third chapter specifies the requirements gathered, both function and non-functional necessary for the development of the system. The fourth chapter gives the design details of the system and provides figures and tables to assist the reader. The fifth chapter details the usability testing and unit testing and its details. The sixth chapter concludes the document itself and gives the reader an insight into the future work that is required and maybe done by the development team. Appendix A includes the user manual to aid the user in understanding the usage and operation of the system. Appendix B points to the various references and resources used in the documentation and development of the system.

2. Literary Review

2.1 Introduction

Previously NUST-LMS (Learning Management System) is working and all the NUST (National University of Sciences and Technology) colleges are using it. It is a web application. NUST-LMS is working and all the NUST colleges are using it.

Since the previously implemented LMS does not offer students to use all feature, the LMS has to offer. The NUST Book will allow its users to access the learning management system (LMS) using through their phones as well as through web. The application shall help the new students at MCS to understand the relative marking system, system of study, maintain and keep record of attendance and their Student progress of the semester. The goal is to make an intuitive application for the students of MCS so that they can access LMS from their mobiles and use it on the go.

NUST Book is an application that will aid in accessing the LMS. This application shall guide the student's through-out their four-year degree program. NUST Book is an application that will aid in accessing the LMS. This application shall guide the student's through-out their four-year degree program.

The application shall keep track of the students' progress and attendance of the current semester as well as the previous semesters. The application shall provide guidelines to the students on how to improve their results. The application will use web scraping, html parsing to create an intuitive interface for exploring the LMS. The application will access LMS to fetch data required for the application to operate properly and therefore the application requires the use of internet to be fully functional.

2.2 Previous Work

Previously NUST-LMS is working and al the NUST colleges are using it. It is a web application. The NUST Book will allow its users to access the learning management system (LMS) using phones and also computers. The goal is to make an intuitive application for the students of MCS so that they can access LMS from their mobiles and use it on the go.

The NUST Book will allow its users to access the learning management system (LMS) using smart phones. The application shall help the new students at MCS to understand the relative

marking system, system of study, maintain and keep record of attendance and their student progress of the semester. The goal is to make an intuitive application for the students of MCS so that they can access LMS from their mobiles and use it on the go.

2.3 Shortcomings

Previously the NUST-LMS, has a difficult to use interface and lack a lot of features required for the students to use. It does not offer students to use all feature, the LMS has to offer. The NUST Book will allow its users to access the learning management system (LMS) using computers and phones. The application shall help the new students at MCS to understand the relative marking system, system of study, maintain and keep record of attendance and their Student progress of the semester. Our system shall provide the mentioned services which were not the part of previous learnig management system.

2.4 Conclusion

The application shall keep track of the students' progress and attendance of the current semester as well as the previous semesters. The application shall provide guidelines to the students on how to improve their results. The application will use web scraping, html parsing to create an intuitive interface for exploring the LMS. The application will access LMS to fetch data required for the application to operate properly and therefore the application requires the use of internet to be fully functional.

3. System Requirements

3.1 Introduction

This section introduces the requirements and system requirements, use cases and other user requirements for the NUST BOOK LMS. This chapter includes the purpose, intended audience, product scope and overall description of the system,

3.2 Purpose

This section describes the software functional and non-functional requirements for the release of the NUST Book web and mobile application. This document is intended to be used by the members of the project team that will implement and verify the correct functioning of the system.

3.3 Intended Audience and Reading Suggestions

Primary readers for this document are the designers and developers of the NUST Book app, faculty of the MCS.

3.4 Product Scope

The NUST Book will allow its users to access the learning management system (LMS) using computers and phones. The application shall help the new students at MCS to understand the relative marking system, system of study, maintain and keep record of attendance and their Student progress of the semester. The goal is to make an intuitive application for the students of MCS so that they can access LMS from their computers and mobiles and use it on the go.

3.5 Overall Description

The overall description in this subsection of this chapter shall let us know about the product perspective that what this application is going to do in what conditions and how will it do it. It will also tell us about product features, user classes and characteristics, operating environment, design and implementation constrains, user documentation and assumptions taken in developing this system.

3.5.1 Product Perspective

NUST Book is an application that will aid in accessing the LMS. This application shall guide the student's through-out their four-year degree program. NUST Book is an application that will aid in accessing the LMS. This application shall guide the student's through-out their four-year degree program.

The application shall keep track of the students' progress and attendance of the current semester as well as the previous semesters. The application shall provide guidelines to the students on how to improve their results. The application will use web scraping, html parsing to create an intuitive interface for exploring the LMS. The application will access LMS to fetch data required for the application to operate properly and therefore the application requires the use of internet to be fully functional. Figure 2.1 shows the context diagram.

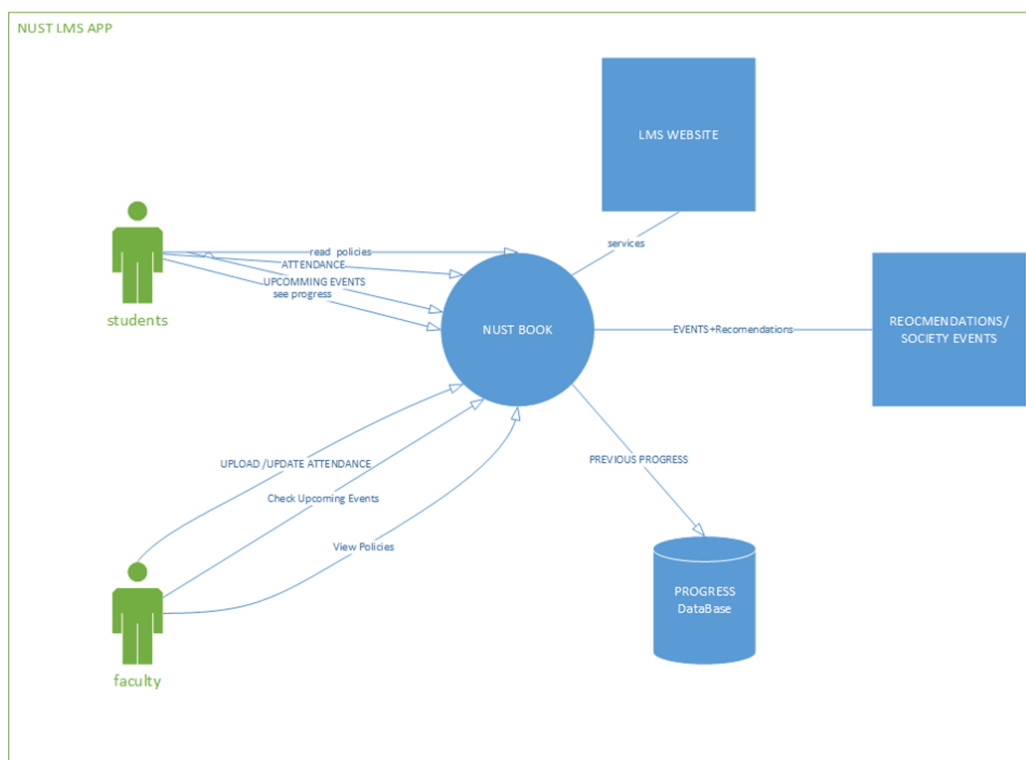


Figure 3.1: Context Diagram

The context view of the system tells us how two users i.e. student and faculty, will use the system. Figure 3.1 shows the context diagram of the system in which the users access the application for different functions for both users and how those requests are forwarded.

3.5.2 Product Features

The NUST Book shall have six subsystems over which the functionality of the application is divided. The main features and subsystems are described as follows:

3.5.2.1 Student progress

This subsystem shall keep track of the progress in the current semester. This shall include progress regarding quizzes, assignments, semester projects, One Hour Test (OHTs), sessional and finals. It shall alert the user about any due assignments or quizzes respectively. It shall include a Grade Point Average (GPA) calculator that shall anticipate the GPA and let the user know about their Student progress and provide guidance to the student to improve their situation.

3.5.2.2 LMS (Learning Management System)

This subsystem shall give free access to the NUST-LMS on the computer and mobile. This shall of course require login prior to use. This subsystem shall provide services provided by the NUST-LMS. The faculty may use this to upload the attendance.

3.5.2.3 Progress Record

This subsystem shall keep record of the all previous semesters. It shall include all the grades of all the subjects, the GPA and the CGPA.

3.5.2.4 Attendance

This subsystem shall use the LMS subsystem to query the attendance of the student. This subsystem shall also provide the services to upload the attendance based upon the user-privilege. This subsystem shall alert the user if the attendance is falling below 75%. It shall also tell the user about how many classes they can skip.

3.5.2.5 Social Events

This subsystem shall query the LMS to get information about the upcoming events in the college. It shall notify the user if any event is close.

3.5.2.6 NUST Policies

This subsystem shall briefly describe the relevant and important NUST policies to the user. The policies shall be categorized into two levels.

3.5.3 User Classes and Characteristics

Following is the list of users who shall be eventually using the system and will interact with the system based on the different access levels granted to each of them.

3.5.3.1 Student

Students shall have access to view their attendance their current courses, their current CGPA, NUST Policies and to download lectures by a proper authentication system. They shall also be able to mark important alerts like upcoming quizzes, assignments and the semester projects etc. and a GPA calculator is also accessible.

3.5.3.2 Faculty

Faculty members shall have the access to upload attendances, quizzes and assignment marks, upload and download lectures. They shall also be able to access upcoming events and NUST Policies

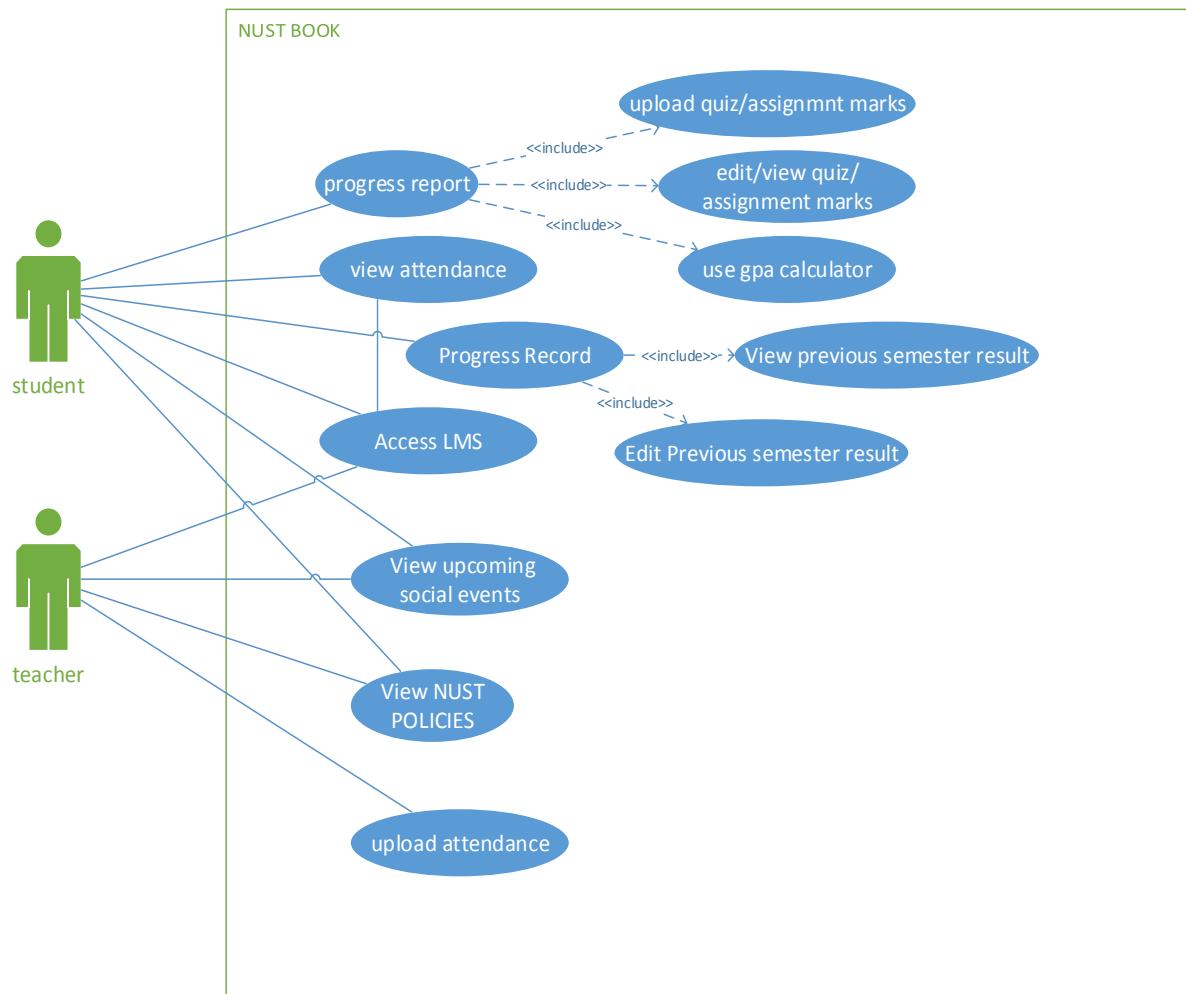


Figure 3.2: Use Case Diagram

3.5.4 Operating Environment

The operating environment shall tell us how the system is going to work efficiently under conditions and environment provided. These include:

3.5.4.1 The application shall operate on the Windows OS and android phones.

3.5.4.2 It shall run on the Windows OS, Android OS version 4.1.2(ICS) and later versions of Android OS.

3.5.4.3 The application is dependent on NUST-LMS only; otherwise it is independent of any other software.

3.5.5 Design and Implementation Constraints

Design and implementation constraints shall tell us that in what conditions the system is probably going to crash or not work properly. So we should have to consider these constraints in order to have correct functioning of the system. These include:

3.5.5.1 The application shall be developed for Windows and android phones only.

3.5.5.2 The application shall require internet for accessing LMS.

3.5.5.3 The application code shall be written in Java.

The Internet connection is a constraint for the product because the application fetches data from the Internet, it is crucial that there is an Internet connection for the application to function.

3.5.6 User Documentation

The system shall have built in help. In addition to this a user manual shall be provided in order to make the users familiar with the system.

3.5.7 Assumptions and Dependencies

The assumptions taken under which the system is assumed to work perfectly are mentioned here. It is very important to consider these assumptions and dependencies for the correct functioning of the system.

3.5.7.1 It is assumed that the user shall have a working internet connection on their device.

3.5.7.2 The users must have basic knowledge of English Language.

3.5.7.3 The user must have the knowledge of using Windows OS and android phones.

3.5.7.4 The system is dependent upon LMS website.

3.6 External Interface Requirements

External interface requirements cover the user interfaces, hardware interfaces, software interfaces, and communication interfaces. The interface requirements for all these interfaces are very important for the correct designing and developing a user friendly environment for the user of this system.

3.6.1 User Interfaces

The system shall have a main menu that shall six items. All the items shall have icons with text below them. The items shall be as follows:

3.6.1.1 Your Progress

3.6.1.1.1 On main screen it shall list all the semester subjects

3.6.1.1.2 On touching a semester subject it shall open a new page with two tabs:

3.6.1.1.3 First tab will be score tab; it shall display subject score, quiz total etc. and it shall also have a button at the bottom that shall open a message box displaying the suggestions on improving and maintaining the subject score.

3.6.1.1.4 Second tab will be pending quizzes/assignments

3.6.1.1.5 Second tab shall have a button with a + sign that shall allow for adding a new quiz or assignment that has occurred.

3.6.1.2 LMS (Learning Management System)

3.6.1.2.1 On touching the LMS a login screen shall appear

3.6.1.2.2 After login the user shall be presented with the LMS

3.6.1.2.3 The LMS shall have a left to right sliding menu

3.6.1.2.4 The sliding menu shall have user name, course etc. and logout options

3.6.1.3 Progress Record

3.6.1.3.1 The progress record screen shall list all the previous semesters and at the bottom the CGPA will be written

3.6.1.3.2 Upon touching the semester it shall list the semester score, namely the subject grades etc.

3.6.1.4 Attendance

3.6.1.4.1 On the main screen it shall display all the current semester subjects.

3.6.1.4.2 Upon touching a subject it shall display the subject's attendance, the attendance percentage and the number of classes that can be skipped without dropping the attendance below 75%.

3.6.1.5 Upcoming Events

3.6.1.5.1 On the main screen it shall display all the events that haven't occurred

3.6.1.5.2 Once an event is near, the app shall send a notification that shall be displayed in the phone's default notification drag down menu.

3.6.1.6 NUST Policies

3.6.1.6.1 On the main screen all the main categories for the NUST Policies shall be listed and upon touching a category a new page shall open with details about that category.

3.6.2 Hardware Interfaces

This app is to be an Windows and android phone application, and as such, will be designed to interface with the hardware present on the iPhone and Android phones. In theory the application will be able to run by other devices that can emulate the Android and Windows OS, but this will not be a consideration during design.

As this is a mobile application, it will be using the cellular network or Wi-Fi to connect to the Internet, which will allow it to communicate with the database servers. This means that it will be using the infrastructure, be it wireless communication points or physical lines, of the network in order to perform properly. There will have to be some sort of error checking for if the network is down or inaccessible.

3.6.3 Software Interfaces

The application is developed in Windows and android environment and it will use all the built in Windows and Android libraries and will also make the use of HTML parser java library. It will use SQL-Lite as its database backend.

The Attendance subsystem shall be dependent on NUST-LMS where user will be authenticated by its LMS username and password after which user will be able to access information according to its privileges.

3.6.4 Communications Interface

The only form outside communication that the app is intended to do is to connect to the LMS- website. It will link to web pages on the LMS-website via HTTPS.

3.7 System Features

Functional requirements define the fundamental actions that system must perform. The functional requirements for the system are divided into six main categories Progress report, LMS, Record Book, Attendance, Upcoming events and NUST Policies.

3.7.1 Student Progress

- 3.7.1.1 This subsystem shall keep the students study record of the current semester provided by the student.
- 3.7.1.2 The Progress Record subsystem shall allow the user to edit the result of a semester.
- 3.7.1.3 The Progress Record subsystem shall allow the student user to enter marks/Grade for a particular subject.
- 3.7.1.4 The Progress Record subsystem shall allow the student user to view marks/Grade for a particular subject.
- 3.7.1.5 The Progress Record subsystem shall allow the student user to edit marks/Grade for a particular subject.
- 3.7.1.6 The Progress Record subsystem shall allow the faculty user to enter marks for assignments/quizzes.
- 3.7.1.7 The Progress Record subsystem shall allow the faculty user to edit marks for assignments/quizzes.
- 3.7.1.8 The Progress Record subsystem shall allow the student user to view marks for assignments/quizzes.
- 3.7.1.9 The Progress Record subsystem shall allow the faculty user to enter the marks for semester projects.
- 3.7.1.10 The Progress Record subsystem shall allow the faculty user to edit the marks for semester projects.
- 3.7.1.11 The Progress Record subsystem shall include a GPA calculator.
- 3.7.1.12 The Progress Record subsystem shall allow the faculty user to upload events like upcoming quizzes and assignment.

3.7.1.13 The Progress Record subsystem shall alert the student user of any upcoming quizzes and assignment.

3.7.1.14 The Progress Record subsystem shall allow the faculty user to guide the student user in improving the student's current academic situation.

3.7.1.15 The Progress Record subsystem shall allow the user to view his remaining credit hours.

3.7.2 LMS

3.7.2.1 The subsystem shall let the student/faculty user full access to NUST-LMS. The user can perform all the functions that he can do with the LMS accessed from the browser.

3.7.2.2 The subsystem shall have a sliding menu that can be opened by a left to right slide of finger on the screen.

3.7.3 Record Book

3.7.3.1 The Record Book subsystem shall allow the student user to view his GPA of previous semesters

3.7.3.2 The Record Book subsystem shall allow the student user to view his CGPA before and after the improvements in the summer semester (if any).

3.7.3.3 The Record Book subsystem shall allow the student user to view his marks in previous quizzes and assignments.

3.7.3.4 The Record Book subsystem shall allow the student user to view his grades in his previous courses.

3.7.3.5 The Record Book subsystem shall allow the student user to view his completed credit hours.

3.7.4 Attendance

3.7.4.1 The Attendance subsystem shall authenticate the student/faculty user on the basis of his LMS-NUST username and password.

3.7.4.2 The Attendance subsystem shall keep track of the student user's attendance of the current semester.

3.7.4.3 The Attendance subsystem shall allow the student user to download the attendance correction form from the NUST-LMS

3.7.4.4 The Attendance subsystem shall only be readable i.e. user can only see the attendance not alter it.

3.7.4.5 The Attendance subsystem shall alert the student user via a notification if the attendance falls between 75%-80%.

3.7.4.6 The Attendance subsystem shall inform the student user of how many absentees he can afford.

3.7.4.7 The Attendance subsystem shall have different privilege level for both students and faculty users

3.7.4.8 The Attendance subsystem shall allow the faculty user to upload the attendance of students

3.7.4.9 The Attendance subsystem shall allow the faculty user to alter the attendance of students

3.7.5 Upcoming Events

3.7.5.1 This subsystem shall keep track of the upcoming events in the college via LMS. It will query the events from LMS and the events can also be manually entered.

3.7.5.2 This subsystem shall alert the user via a notification if any new events are close.

3.7.6 NUST Policies

3.7.6.1 The policies subsystem shall let the user view the important policies of NUST.

3.7.6.2 The policies shall be categorized into the different sub-categories. I.e. Policies for Undergraduate students will be different from MS students and faculty

3.7.6.3 The policies will be pasted as such from NUST website

3.7.6.4 The policies shall be explained in a simplified and brief manner.

3.8 Nonfunctional Requirements

The nonfunctional requirements of the system include the performance requirements, safety requirements, security requirements, portability requirements, adaptability, usability and maintainability requirements.

3.8.1 Performance Requirements

Performance requirements define acceptable response times for system functionality. The system shall provide appropriate responses to all the interactions with the system. The system shall load each screen in one second. The system shall return results to a query in three seconds.

The query to attendance shall take five seconds as it would access the NUST-LMS website.

3.8.2 Safety Requirements

There are no safety requirements for NUST Book except for the usual safety requirement for the use of cellphone such as; not to use the phone while you are driving etc.

3.8.3 Security Requirements

Each user shall have to use its LMS username and password in any subsystem of the application where it needs to access NUST-LMS.

3.8.4 Software Quality Attributes

3.8.4.1 Availability

The system's availability is dependent upon internet, if the internet is not available the system shall not be able to access the NUST-LMS and all the functions of app that require the use of LMS shall not be available. However some features of the app are offline such as the progress record, they shall be available offline regardless.

3.8.4.2 Portability

The system shall be developed only for Windows and Android OS (4.1.2). The system shall run on all the devices running these OSs. Later on, ports for various other devices such as, Windows Phone, iOS shall also be created.

3.8.4.3 Usability

The system has a very intuitive interface and shall be very easy to use. The interface shall be quite simple and the learning curve shall be smooth.

3.8.4.4 Adaptability

The system shall be adaptable to the needs of user. The interface shall be customizable to some degree. There shall be no special configuration needed for the system to use.

3.8.4.5 Maintainability

The project team shall release a new version of the app every six months. The new version shall fix the bugs found in the current versions.

3.9 Conclusion

The NUST Book was finally able to allow its users to access the learning management system (LMS) using computers and phones. The application shall help the new students at MCS to understand the relative marking system, system of study, maintain and keep record of attendance and their Student progress of the semester. The goal, to make an intuitive application for the students of MCS so that they can access LMS from their computers and mobiles and use it on the go, was finally achieved.

4. System Design Specifications

4.1 Introduction

This section introduces the design specification and system design specification, use cases and other user design specification for the NUST BOOK LMS. This chapter includes Purpose, Product scope, intended audience, references and document overview.

4.2 Purpose

This section describes the software functionality based on the design of the product using different design approaches before implementing it. This document is intended to be used by the members of the project team that will implement and verify the correct functioning of the system.

4.3 Product Scope

The NUST Book will allow its users to access the learning management system (LMS) using computers and smart phones. The application shall help the new students at MCS to understand the relative marking system, system of study, maintain and keep record of attendance and their student progress of the semester. The goal is to make an intuitive application for the students of MCS so that they can access LMS from their mobiles and use it on the go.

4.4 Intended Audience and Reading Suggestions

Primary readers for this document are the developers of the NUST Book app and faculty of the MCS.

4.5 Document Overview

The next chapter of document has described architectural design of Nust-Book System. The high level components and their interactions, suitable architectural pattern and design decisions applied to the whole system.

The final chapter of the System design specification is on component and detailed design. It includes design patterns, Sequence diagrams, class diagrams, activity diagram, state transition diagram, use case diagrams and user interface diagram.

4.6 System Architecture Description

This subsection of this chapter will tell us that which architectural pattern is being used in developing the system and the reasons for it being used. It will also describe the pattern that is being used according to the structure of the modules.

4.7.1 Overview of modules

Identification of each module of the system is done here along with the key descriptions of each module is also written in this subsection of this chapter.

4.7.1.1 Module identification

4.7.1.1.1 Login Module

This is one of the key modules of the system. This is where users are authenticated and given privileges according to the type of the user.

4.7.1.1.2 Attendance Module

This module has two parts, uploading attendance and checking attendance. Student's attendance record is maintained by this subsystem.

4.7.1.1.3 Subject Registration Module

In this module subjects are registered by the student users at the end of each semester for the upcoming semester.

4.7.1.1.4 Grades Module

This module has all the previous data on each subject enrolled by each student of the university. All the grades scored on courses studied in the previous semesters are stored through this module.

4.7.1.1.5 Improvements Module

This module takes care of the courses repeated/improved. It maintains the important information and all previous record for the courses that are opted to be improved.\

4.7.1.1.6 Previous Record Check Module

This module stores all the information of the courses completed, grades scored in those courses, any improvement made in those courses, their improved grades and any extra subject completed.

4.7.1.1.7 Balanced Module

This module stores all the information regarding the credit hours completed and the credit hours left.

4.7.1.1.8 Internship Check/Update Module

This module contains Important updates regarding Monthly internships for the students of NUST.

4.7.1.1.9 Scholarship Check Module

This module stores important information regarding different scholarships offered at NUST and their criteria.

4.7.1.1.10 Event Check Module

This module contains important updates regarding the events that are going to take place at NUST and is constantly updated.

4.7.1.2 Architecture patternen

The architectural pattern used for this system is **MVC (Model View Controller)**.

MVC style separates presentation and interaction from the system data. The system is structured into three logical components that interact with each other.

The **Model** component- Manages the system data and associated operations on that data. The **View** component- Defines and manages how data is presented to the user.

The **Controller** component- Manages user interaction and passes these interactions to the View and Model.

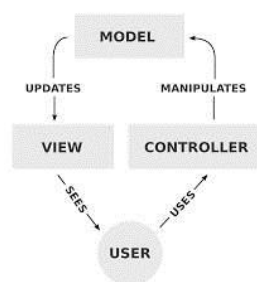


Figure 4.1: MVC

Figure 2.0 shows the component diagram of the MVC pattern. We will use MVC style for the NUST-Book because there are multiple ways to view and interact with data. Also used when the future requirements for interaction and presentation of data are unknown. In some software systems, the code between process logic and interface are mixed. This will reduce the modularity of application and make the system more difficult to maintain. To avoid this problem, we have decided to use MVC architectural style to separate the application logic with the interface. The main advantage of this style allows the data to change independently of its representation and vice versa. Support presentation of the same data in different ways with changes made in one representation shown all of them.

4.7.1.3 Component Identification

Identification of each component of the system is done here. Each component is first identified, and then its type, purpose, subordinates, dependencies, interfaces, resources, processing and data is described.

4.7.1.3.1 Component Template Description

Table 4.1: Component Template Description

Identification	The unique name for the component and the location of the component in the system.
Type	A module, a subprogram, a form, a data file, a control procedure, a class, etc
Purpose	Function and performance requirements implemented by the design component, including derived requirements. Derived requirements are not explicitly stated in the SRS - but are implied or adjunct to formally stated SDS requirements.
Subordinates	The internal structure of the component, the constituents of the component, and the functional requirements satisfied by each part.
Dependencies	How the component's function and performance relate to other components. How this component is used by other components. The other components that use this component. Interaction details such as timing, interaction conditions (such as order of execution and

	data sharing), and responsibility for creation, duplication, use, storage, and elimination of components.
Interfaces	Detailed description of all external or internal interfaces as well as of any mechanism for communicating through messages, parameters, or common data areas. All error messages and error codes should be identified. All screen formats, interactive messages, and other user interface components (originally defined in the SRS) should be given here.
Resources	<i>A complete description of all resources (hardware or software) external to the component but required to carry out its functions.</i>
Processing	A full description of the functions presented in the Function subsection. Pseudocode can be used to document algorithms, equations, and logic.
Data	For the data internal to the component, describes the representation method, initial values, use, semantics, and format.

The name of the component should be unique to easily identify them individually. For that reason the identification of modules, its types and all other mentioned fields are fully met.

4.7.1.3.2 Database

Table 4.2: Database

Identification	Database
Type	Database/ Data File
Purpose	Database shall be used to store all the data relevant to the system. This includes Students' data like Name, Course, GPA summary faculty's data, all the data about the current events and past events. It shall also have all the data about courses in the college.

Function	This component is used for all types of data storage in the system. Other than this it doesn't do anything.
Subordinates	The database shall have 6 tables. User, Subject, courses, groups, posts, events. The user table shall store info about students and faculty. Subjects table shall have all the information about the subjects in the university whether active or not. The courses shall contain info about the current courses and past courses in the college. The posts table shall contain info about about each post made in the system.
Dependencies	All the other components are dependent on the database. If the database is down no other component shall be able to work properly. All the components are tightly coupled with the database.
Interfaces	The database shall provide interfaces for all the components. This includes piStudent for the Student component, piFaculty for the Faculty component, piGroups for the Group component, piSubjects for the Subject component, piCourses for the Courses component.
Resources	<i>The database doesn't require any resources.</i>
Processing	None
Data	This component is the central data point for the system. The data shall be used in all the components of the system. It shall be stored in tables.

Database shall be used to store all the data relevant to the system. This includes Students' data like Name, Course, GPA summary faculty's data, all the data about the current events and past events. It shall also have all the data about courses in the college.

4.7.1.3.3 Login

Table 4.3: Login

Identification	Login
Type	Class/Form
Purpose	To authenticate the user
Function	This component shall authenticate the user to the system. It shall also tell the system what is the status of this user i.e. Faculty or Student.
Subordinates	Consists of three things: username field, password field and a submit button which shall submit the data input by the user.
Dependencies	No other component shall provide any service if the Login fails. All the subsystems are dependent on it.
Interfaces	Provides two external interfaces for the student and faculty.
Resources	<i>Access to the User table from the database component is required.</i>
Processing	Access the database to find the username provided by the user and match the password against it. If the combination is correct, check whether the user is Student or Faculty. Based upon the decision, redirect the user to their respective page.
Data	The data that is used in this component is related to authentication. It is not internal and is accessed from the database component.

This component shall authenticate the user to the system. It shall also tell the system what is the status of this user i.e. Faculty or Student. The main reason is off course to authenticate users.

4.7.1.3.4 Student

Table 4.4: Student

Identification	Student
Type	Class/Form
Purpose	This component shall be used to represent a student user. It shall contain all the necessary data and functions that student needs to perform.
Function	Through this component the student user shall interact with the system. It shall provide all the necessary interfaces for the student to interact with the system.
Subordinates	It shall inherit the internal structure of the User component. Other than that it shall add following constituents to the component: Student's course, Current subjects, results of each previously studied subjects, GPAs, CGPA, previously studied subjects.
Dependencies	It depends on the Faculty component for subject access. It also depends on the message, post, group and event components for creating messages, posts in groups, access to groups, creating groups, creating and viewing events.
Interfaces	The student component shall have a different interface than Faculty component. It has a require interface that shall be used to communicate to the faculty component. All other interfaces shall be inherited from the User component.
Resources	<i>Database.</i>

Processing	Provide an interface to the student user for interacting with the system. This interface shall be a webpage and shall be constructed in HTML/CSS. The interface shall provide all the functions like sending a message, creating a post, accessing subjects etc.
Data	Each student user shall have their own data. This contains their Name, username, password, course, joining date, ID. The initial values for these shall be NULL. All this data shall be stored in the database.

This component shall be used to represent a student user. It shall contain all the necessary data and functions that student needs to perform. Through this component the student user shall interact with the system. It shall provide all the necessary interfaces for the student to interact with the system.

4.7.1.3.5 User

Table 4.5: User

Identification	User
Type	Class/Form
Purpose	To provide all the interfaces required for using the system. Separates the student from faculty.
Function	Through this component the user shall Login into the system. The system shall then decide whether the user belongs to student category or faculty and then based upon the result it shall redirect the user to their page. Through this component the user shall interact with the system. It shall provide all the necessary interfaces for the student/faculty to interact with the system.
Subordinates	Student component and Faculty Component. Student component is intended towards the student user. Faculty component is intended towards the faculty user.

	This component consists of: User's name, Status, Messages, Posts created, groups created, member of any groups, Notifications. It shall create a wall for the user where the current user activity shall be available.
Dependencies	Depends on the database for data access. Depends on User, group, events, messages, login for providing services to the user.
Interfaces	It only provides the interface for Login. Based upon the login credentials the User component shall direct the user towards Student or Faculty page.
Resources	<i>Required database to be functional. No hardware resources required.</i>
Processing	Provide an interface to the user for Login. This interface shall be a webpage and shall be constructed in HTML/CSS. Once the user has typed in their credentials, this component shall query the system and match the credentials in the database. If it finds a match the user is authenticated. It shall then check whether the user is a Faculty or Student.
Data	No internal data. All the data required by this component to be functional resides in the database.

To provide all the interfaces required for using the system. It separates the student from faculty.

4.7.1.3.6 Faculty

Table 4.6: Faculty

Identification	Faculty
Type	Class/Form
Purpose	This component shall be used to represent a faculty user. It shall contain all the necessary data, interfaces and functions that faculty needs to perform.
Function	Through this component the faculty user shall interact with the system. It shall provide all the necessary interfaces for the faculty to interact with the system.
Subordinates	It shall inherit the internal structure of the User component. Other than that it shall add following constituents to the component: Faculty's ID, faculty's rank, current subjects teaching, previously taught subjects, achievements.
Dependencies	It depends on the Subject components' interface for creating subjects which the students shall use. It also depends on the message, post, group and event components for creating messages, posts in groups, access to groups, creating groups, creating and viewing events.
Interfaces	The faculty component shall have a different user interface than student component. It has a require subject interface that is required for creating the subjects that the students can access. All other interfaces shall be inherited from the User component.
Resources	<i>Database.</i>
Processing	Provide an interface to the faculty user for interacting with the system. This interface shall be a webpage and shall be constructed in HTML/CSS. The interface shall provide all the functions like sending a message, creating a post, accessing subjects etc.
Data	Each faculty user shall have their own data. This is not internal to the component rather this data resides in the database and shall be available whenever an authenticated user requires it.

This component shall be used to represent a faculty user. It shall contain all the necessary data, interfaces and functions that faculty needs to perform

4.7.1.3.7 Subject

Table 4.7: Subject

Identification	Subject
Type	Class/Form
Purpose	This component shall be used to manage all the subjects in the system.
Function	This component shall be used to manage all the subjects in the system. It shall provide access to the faculty member which can create and manage a subject. The subject can then be accessed by students which are enrolled in that subject. It shall also provide all the necessary functions for managing the subject like deleting subject, removing a student, accepting enrollments.
Subordinates	It shall consist of: Subject ID, Name, Number of students, Subjects' Instructor, total credits.
Dependencies	Depends on the database for data access. The subject in itself doesn't do anything.
Interfaces	Provides an interface for the faculty to use. The faculty shall use this interface to create and manage a subject.
Resources	<i>Database.</i>
Processing	Provides interface to the faculty member which the faculty member shall use to create and manage the subject. There shall be no central point for managing all the subjects. The subjects shall be maintained

	by their respective Faculty or creators. The faculty shall create a subject and then open enrollment to the students. If a student enrolls, the faculty shall be prompted to accept or reject the enrollment request. The faculty can also remove a student from the subject.
Data	Each subject shall contain info. Pertaining to the subject itself. This includes Subject ID, Assigned Instructor, Total Students, Total Credits. It shall be viewable by the subject creator. The data doesn't reside inside the component and shall be accessed upon request from the database.

This component shall be used to manage all the subjects in the system. It shall provide access to the faculty member which can create and manage a subject. The subject can then be accessed by students which are enrolled in that subject. It shall also provide all the necessary functions for managing the subject like deleting subject, removing a student, accepting enrollments

4.7.1.3.8 Message

Table 4.8: Message

Identification	Message
Type	Class/Form
Purpose	This component shall be used to manage all the communication in the system.
Function	This component shall be used to manage all the communication in the system. It shall provide interfaces to the faculty and student users to send and receive messages. It shall only be used for communication internal to the system.
Subordinates	It shall consist of: Message subject, Message Text, Recipient and

	Sender.
Dependencies	Depends on database for sending and receiving messages.
Interfaces	Provides an interface for the user to use. The user shall use this interface to send and receive messages.
Resources	<i>Database.</i>
Processing	It shall provide interfaces to the faculty and student users to send and receive messages. When a user sends a message to another user, the message shall be inserted in the corresponding user's message table in the database. When the recipient user opens their inbox, the message component shall check the database for any new messages. If any message is there it shall display it to the user.
Data	Each user shall have their own Message Utility. The data shall be split in two categories: sent messages and inbox. Initially they shall be empty.

This component shall be used to manage all the communication in the system. It shall provide interfaces to the faculty and student users to send and receive messages. It shall only be used for communication internal to the system.

4.7.1.3.9 Post

Table 4.9: Post

Identification	Post
Type	Class/Form
Purpose	This component shall be used to create and manage posts.
Function	This component shall be used to create and manage posts. These include posts made in the groups, posts made by teachers in the groups, on the subject pages. Posts can have attachments. File

	uploading is part of posts.
Subordinates	It shall consist of: Post Content, Post Creator, Attachments, Images.
Dependencies	Depends on database.
Interfaces	Provides an interface for the user to create and edit posts.
Resources	<i>Database.</i>
Processing	It shall provide interfaces to the faculty and student users to create and edit posts. Posts can contain images and files. Faculty shall posts lectures and other news in form of posts. When a post is made, the post shall be added to the recipient page's Post Column in the database. The page shall then appear on that page upon page refresh.
Data	Post shall contain three data fields. Post Creator, time the post was created and Post Content. All of these shall be publicly available and shall be stored in the database.

This component shall be used to create and manage posts. These include posts made in the groups, posts made by teachers in the groups, on the subject pages. Posts can have attachments. File uploading is part of posts.

4.7.1.3.10 Group

Table 4.10: Group

Identification	Group
Type	Class/Form
Purpose	This component shall be used to create and manage groups

Function	This component shall be used to create and manage groups. Groups can be created by anyone faculty or student. This component shall provide the following functions: creation of groups, editing of groups, removing moderator, adding members to group.
Subordinates	Consists of: Group Name, Group ID, Posts.
Dependencies	Depends on database and post component.
Interfaces	Provides an interface for the user to create and manage groups. It shall require the post interface for creation of posts in the group.
Resources	<i>Database.</i>
Processing	It shall provide interfaces to the faculty and student users to create and manage groups. Groups can be created by both faculty and students. Whenever a group is created, the new groups' id and name is added to the database. Posts are the main element inside the group. Each group shall have members. Members can be added and removed by the group-moderators. Group moderators are set by the group-creator.
Data	Group shall contain the following data fields. Group Members, Group Content which contains all the post, Group Moderators, Group privacy. Initially Group Members, Group Content and Moderators shall be NULL and the Group Privacy shall be set to Public.

This component shall be used to create and manage groups. Groups can be created by anyone faculty or student. This component shall provide the following functions: creation of groups, editing of groups, removing moderator, adding members to group.

4.7.1.3.11 Events

Table 4.11: Events

Identification	Events
----------------	--------

Type	Class/Form
Purpose	This component shall be used to create and manage Events
Function	<p>This component shall be used to create and manage Events. Events can be created by both student and faculty. It shall provide the following functions: Creation of events, Editing an event, inviting people to event.</p> <p>The system shall manage an Event List of all the public events which shall be used to create an event calendar.</p>
Subordinates	It shall consist of: Event ID, Event Name, Event Organiser, Event Privacy.
Dependencies	Depends on database, Post component and Message Component.
Interfaces	Provides an interface for the user to create, edit and manage events.
Resources	<i>Database.</i>
Processing	<p>It shall provide interfaces to the faculty and student users to create, edit and manage events. Event organizer can invite people to his event. Whenever an organizer sends an invitation, the component shall require the use of Message component to generate an automatic message which shall be sent to the invitee. Similarly if the event date is changed the system shall automatically alert all the people invited to the event of the new date.</p> <p>If a new event is created and the event's privacy is set to Public, the system shall</p>
Data	Each event shall have its own private data. This includes event name, event content i.e. all the posts created on the event page, invitee list, event organisers, event date. The event creator shall be prompted to set initial values for these fields when he creates the event. All this data shall be stored in the database.

This component shall be used to create and manage Events. Events can be created by both student and faculty. It shall provide the following functions: Creation of events, Editing an event, inviting people to event.

The system shall manage an Event List of all the public events which shall be used to create an event calendar.

4.7.2 Structure and relationship

Diagrams have shown gives a pictorial view of how the system works. It comprises of Class diagram, Activity diagram, State transition diagram and Sequence diagrams.

4.7.2.1 Class Diagram

The figure shows the class diagram of the system, showing interclass dependencies. Two main users are shown I.e. Student and Faculty along with other classes like Subject, Events, and Groups etc.

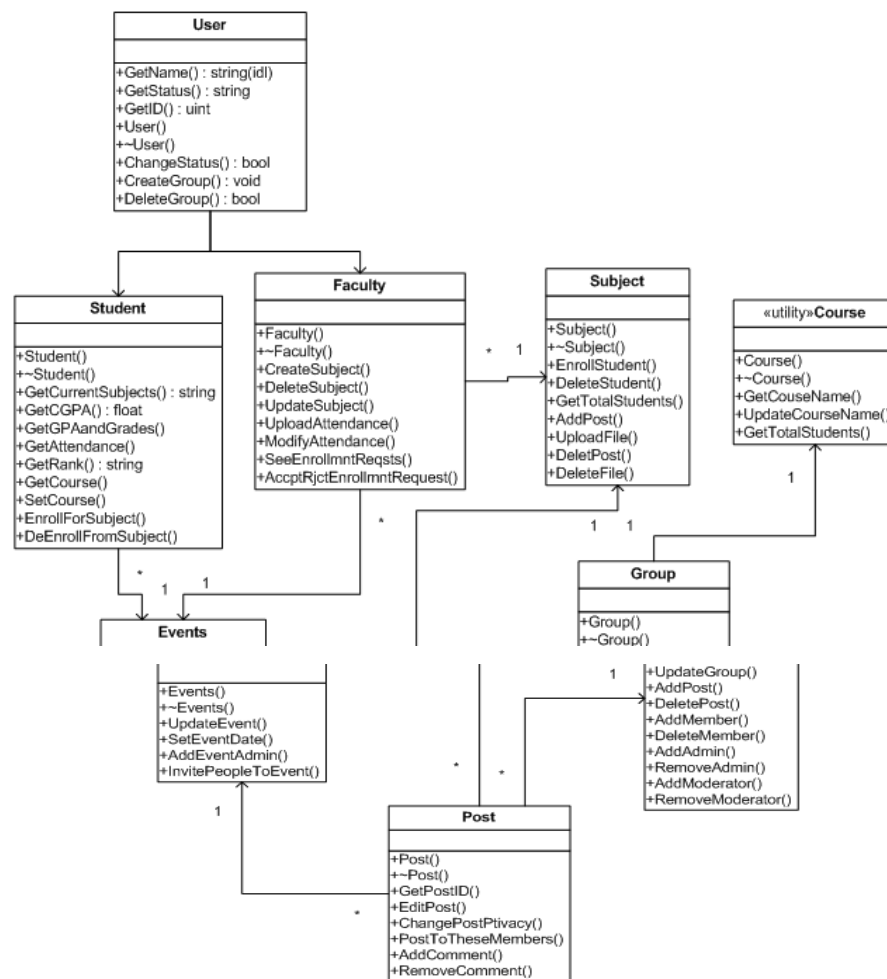


Figure 1.2: Class Diagram

4.7.2.2 Sequence Diagram

Figure 2.2 shows the sequence diagram for Login module.

4.7.2.3 Sequence Diagram for Component 1(Login)

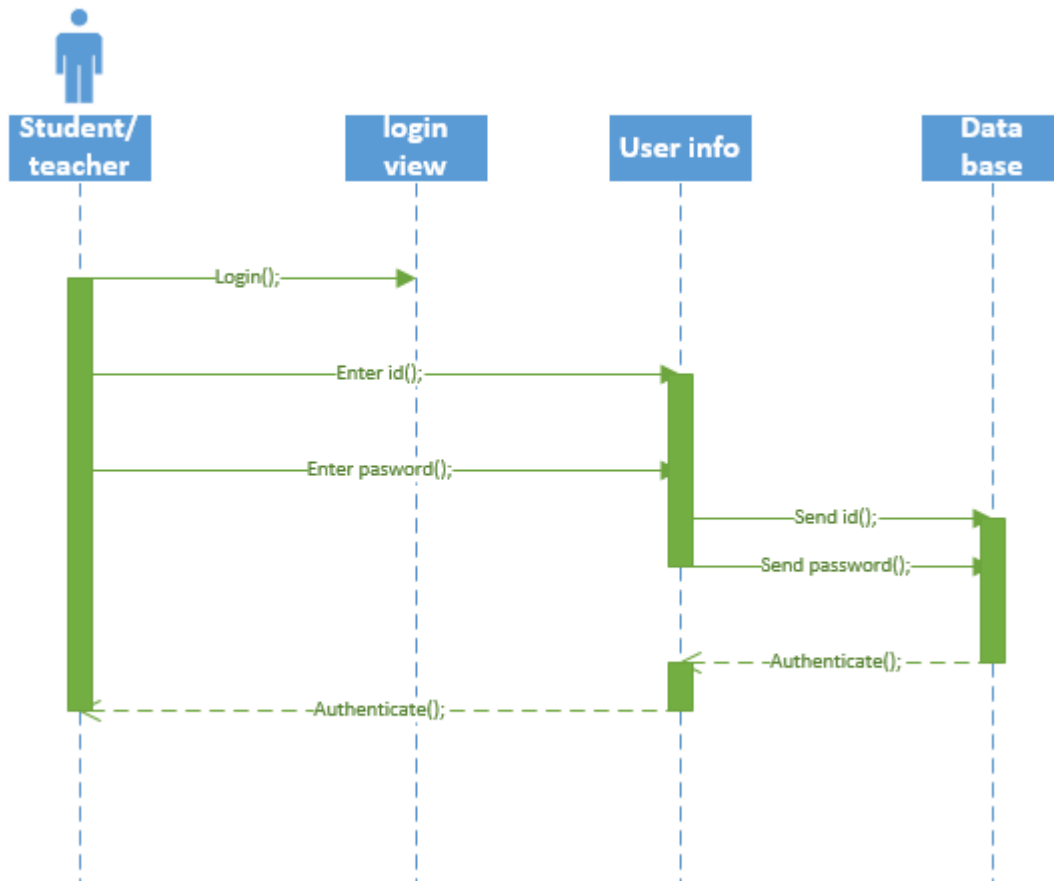


Figure 4.3: Login Sequence

Login module helps to authenticate both users. This is one of the key modules of the system. This is where users are authenticated and given privileges according to the type of the user.

4.7.2.4 Sequence Diagram for Component 2(Attendance)

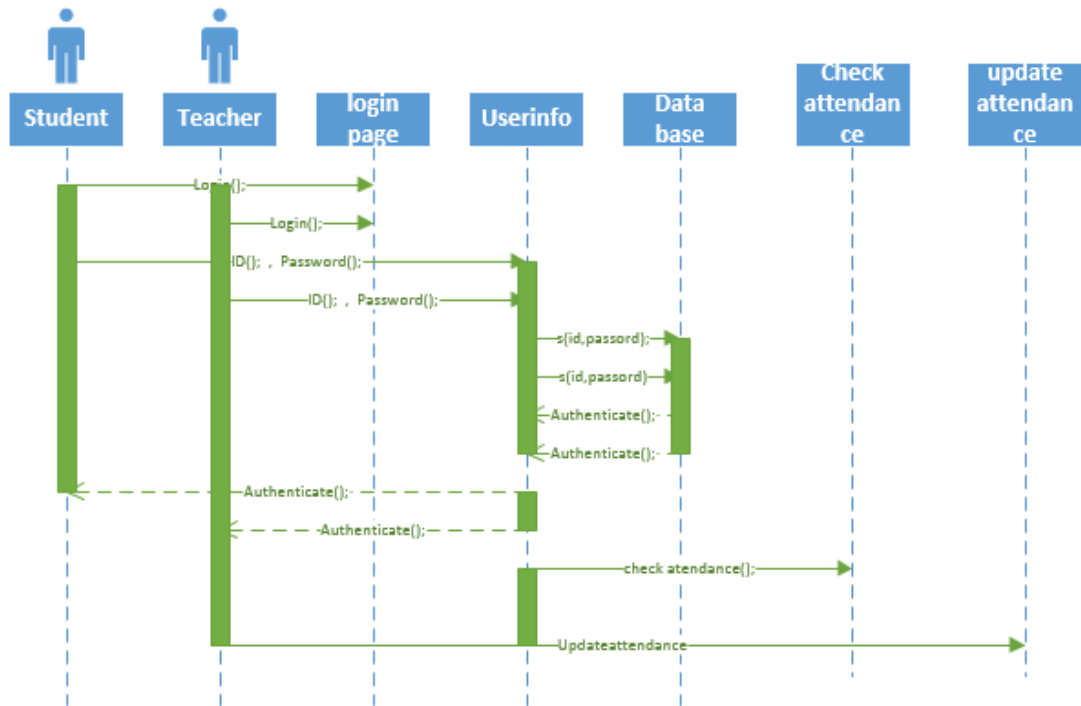


Figure 4.4: Attendance Sequence

This module has two parts, uploading attendance and checking attendance. Student's attendance record is maintained by this subsystem. It requires login prior to use both functions.

4.7.2.5 Sequence Diagram for Component 3(Grades)

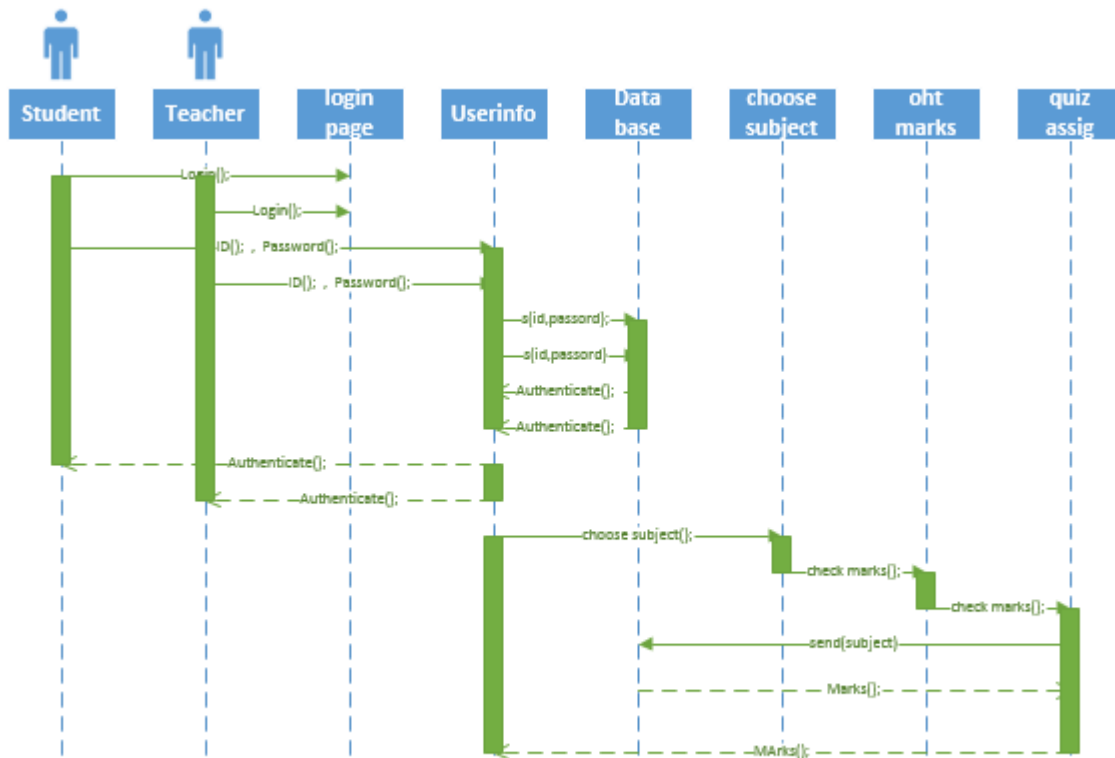


Figure 4.5: Grades Sequence

This module has all the previous data on each subject enrolled by each student of the university. All the grades scored on courses studied in the previous semesters are stored through this module.

4.7.2.6 Sequence Diagram for Component 4(Subject Registration)

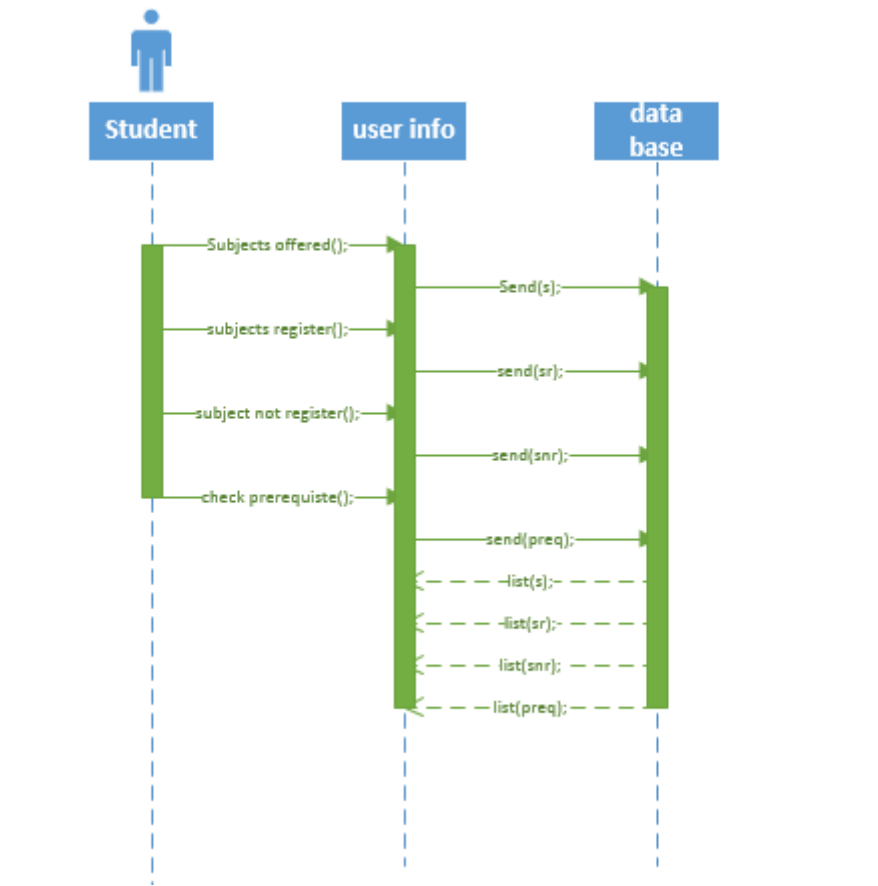


Figure 4.6: Subject Registration Sequence

In this module subjects are registered by the student users at the end of each semester for the upcoming semester. The data of each student is then maintained in a database distinctively.

4.7.2.7 Sequence Diagram for Component 5 (Improvements)

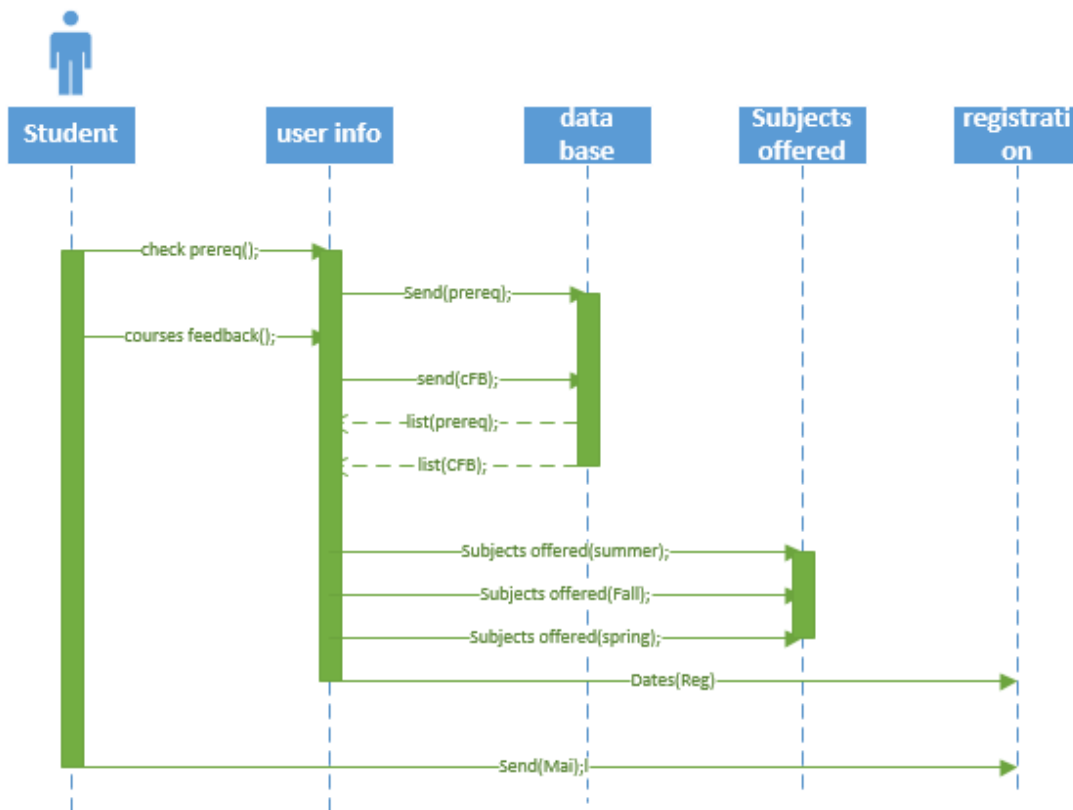


Figure 4.7: Improvements Sequence

This module takes care of the courses repeated/improved. It maintains the important information and all previous record for the courses that are opted to be improved.

4.7.2.8 Sequence Diagram for Component 6(Balanced)

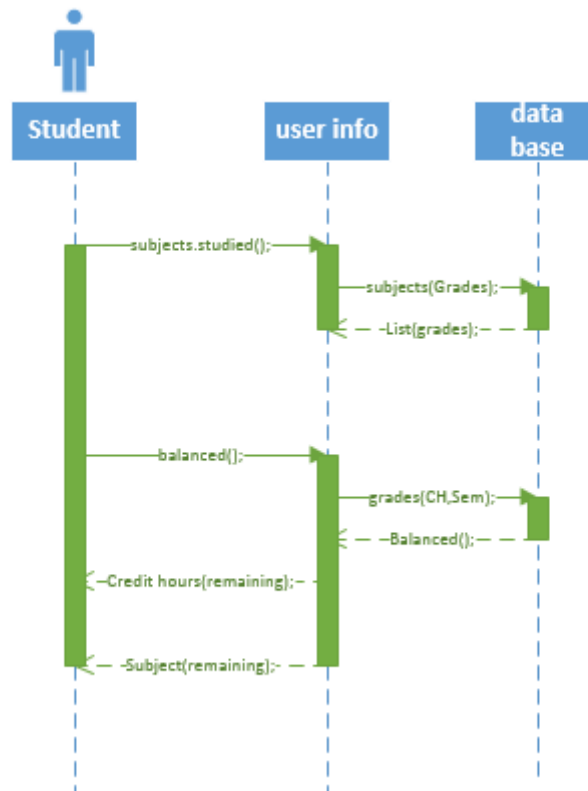


Figure 4.8: Balanced Sequence

This module stores all the information of the courses completed, grades scored in those courses, any improvement made in those courses, their improved grades and any extra subject completed.

4.7.2.9 Sequence Diagram for Component 7(Scholarships and Internships)

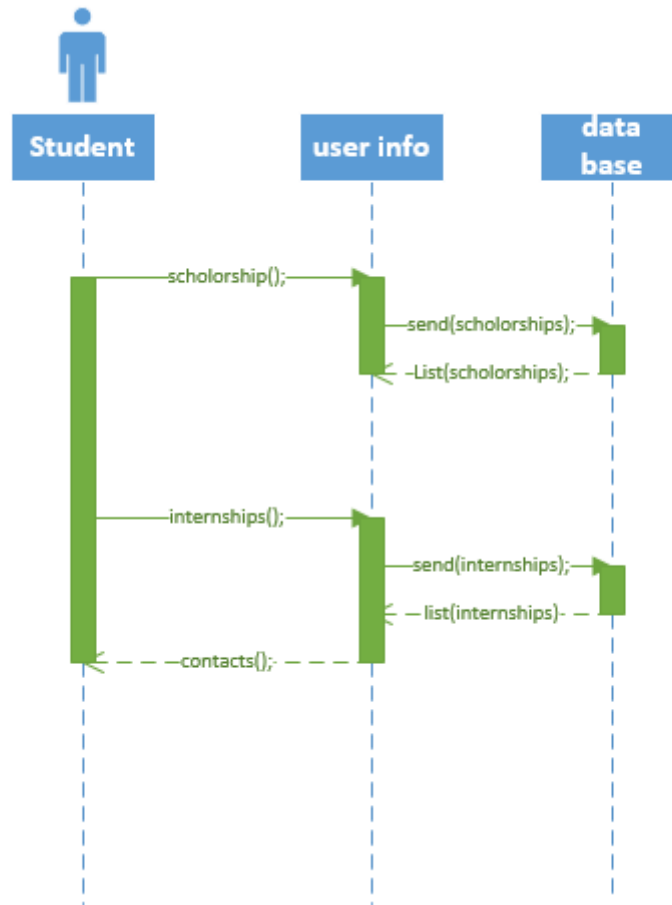


Figure 4.9: Internship/Scholarship Sequence

This module contains important updates regarding Monthly internships for the students of NUST.

4.7.2.10 Sequence Diagram for Component 8 (Events)

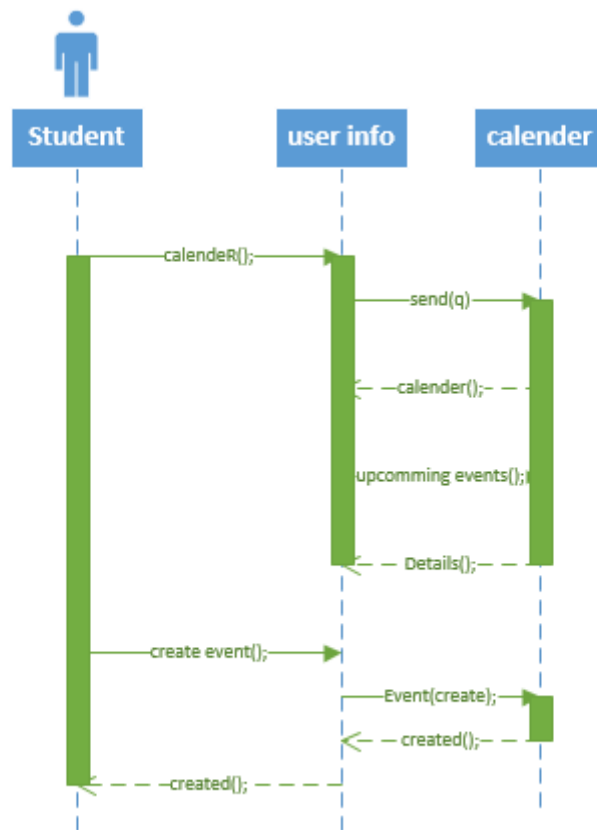


Figure 4.10: Events Sequence

This module contains important updates regarding the events that are going to take place at NUST and is constantly updated. The calendar highlights all the important events that are going to take place in coming days and weeks.

4.7.3 Activity Diagram

Figure 2.10 shows the activity diagram of the system showing the flow of activities by both users i.e. Student and Faculty. Both users have different pages with different set of activities. The login module is required prior to use both users' activities.

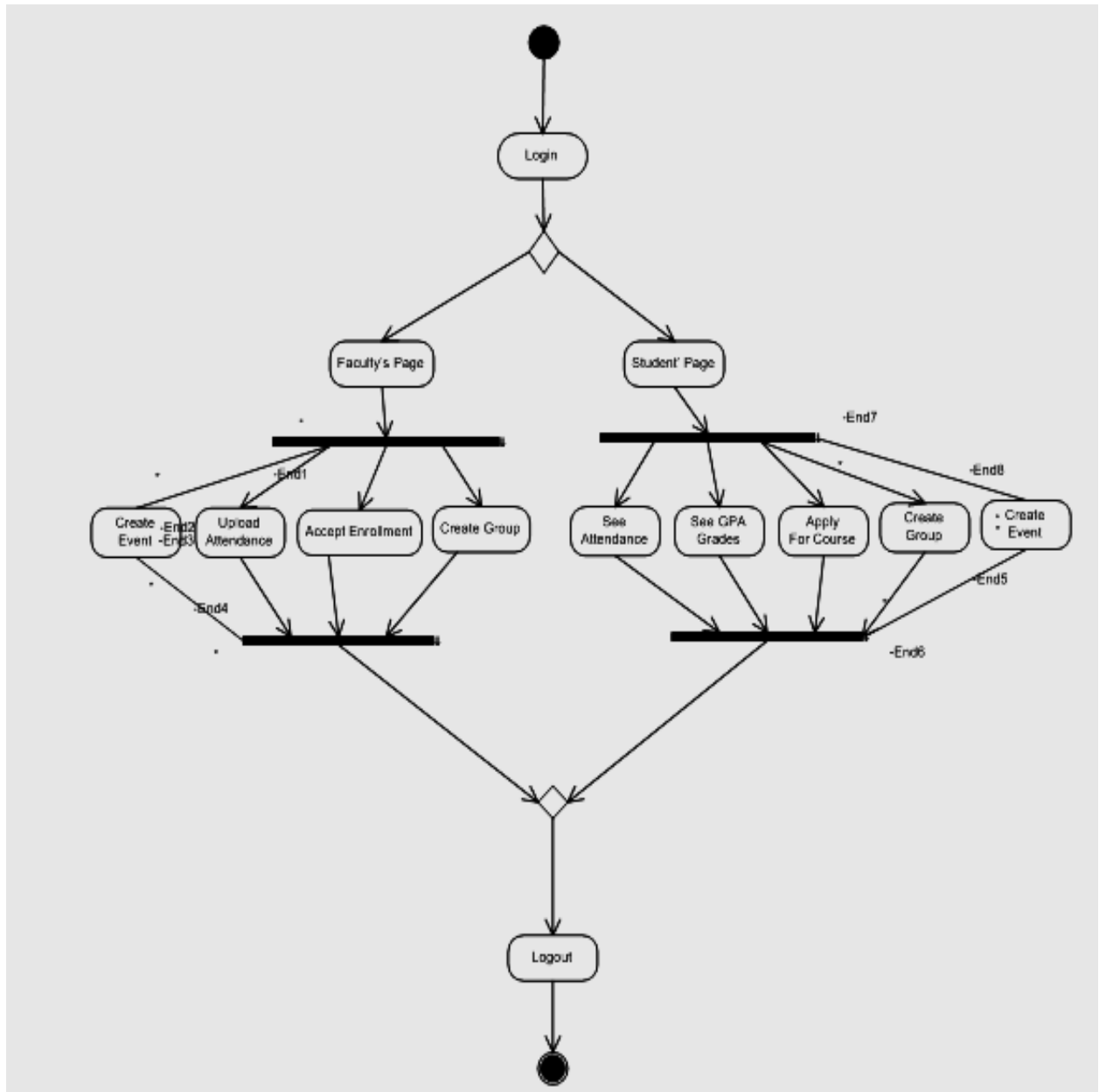


Figure 4.11: Activity Diagram

4.7.4 State Machine Diagram

Figure 2.11 shows the state machine diagram of the system showing the transitions of each activity. Some modules are to be executed first in order to to execute other functions.

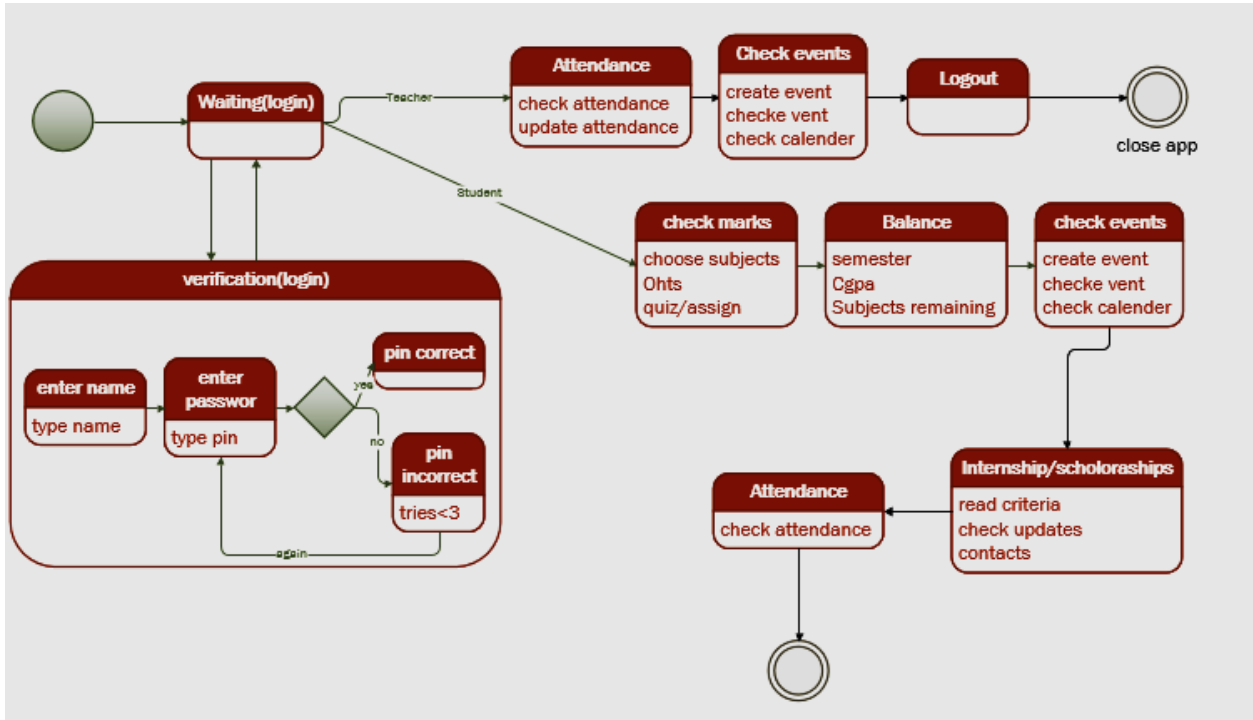


Figure 4.12: State Machine Diagram

4.7.5 Use Case Diagram

The use case diagrams shown below shows the activities and fuctions performed by the users. i.e. Student and Faculty. Use case diagrams include Main use case diagram and use case diagrams of all other modules separately. Also it includes the fully dressed use cases of all the modules.

These diagrams differentiate what user can perform what activities according to the privileges of both users.

4.7.5.1 MAIN USE CASE DIAGRAM

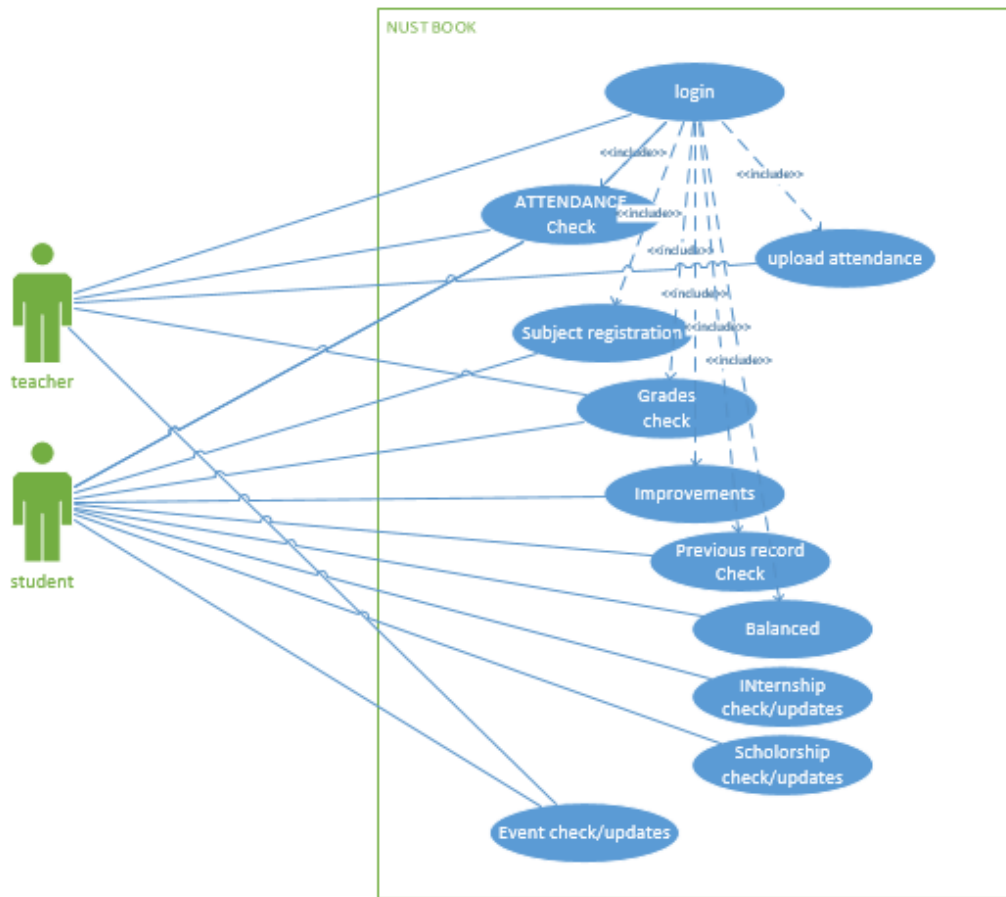


Figure 4.13: Main Use Case Diagram

4.7.5.2 FULLY DRESSED USECASE

This feature shows the fully dressed use case of login module. This feature of Login shall authenticate both the student and the teacher with a username and password. The username and password is authenticated through a database maintained for the system. Each user have different levels of privileges. This is one of the key modules of the system. This is where users are authenticated and given privileges according to the type of the user.

Table 4.12: Login

Use Case ID:	01
Use Case Name:	Login
Actor:	Teacher, Student
Description:	This feature of Login shall authenticate both the student and the teacher with a username and password. The username and password is authenticated through a database maintained for the

	system. Each user have different levels of privileges.
Pre-conditions:	The user must have switched on the application
Post-conditions:	The user will have a list of actions which he/she can choose i.e. Attendance Check, Attendance Upload, Subject Registration, Check Grades, Check Previous Record and Check Balance Studies.
Priority :	This feature is given high priority as it is the base of choosing other actions.
Frequency Of Use:	Only one time at the beginning of the application.
Normal Course of Events:	The user starts the application and authenticates himself/herself.
Alternative Course:	The user selects the options like Event/Updates checks, Internship and Scholarships checks, or user selects to exit the application in which case the application will close down.
Special Requirements:	
Assumptions:	
Notes and Issues:	

This is the fully dressed use case of Attendance module. This feature shall allow students to check their attendance in each registered course whereas teacher can check the attendance of any student. This module has two parts, uploading attendance and checking attendance. Student's attendance record is maintained by this subsystem.

Table 4.13: Attendance Check

Use Case ID:	02
Use Case Name:	Attendance Check
Actor:	Student, Teacher
Description:	This feature shall allow students to check their attendance in each registered course whereas teacher can check the attendance of any student.
Pre-conditions:	The user must have opened the application and must have authenticated him/her-self first.
Post-conditions:	The student would be able to check their attendance whereas teachers would be able to check attendance of any student in his/her class.

Priority :	This feature has a low priority as it is an independent feature and no other feature depends on this feature.
Frequency Of Use:	This feature can be used as many times as user wishes to use. Thereby this feature has a frequency of use.
Normal Course of Events:	<ol style="list-style-type: none"> 1. The user selects to check attendance. 2. If the user is student, the system will display that student's attendance of each course. 3. If the user is teacher, the system will display the attendance of all the students registered to that subject that teacher is teaching.
Alternative Course:	<p>If the user is Teacher :</p> <ol style="list-style-type: none"> 1. The user selects the option of upload attendance. 2. The user selects the option of Grades Check <p>If the user is Student :</p> <ol style="list-style-type: none"> 1. The user selects the option of subject registration 2. The user selects the option of Grades Check 3. The user selects the option of Improvements 4. The user selects the option of Previous records 5. The user selects the option of Balanced 6. The user selects the option of Internships/Scholarships offered and Event checks/Updates.
Special Requirements:	
Assumptions:	
Notes and Issues:	

This feature of Upload attendance shall allow the teacher to upload the students' attendance taken in class. This feature of Upload attendance shall allow the teacher to upload the students' attendance taken in class

Table 4.14: Upload Check

Use Case ID:	03
Use Case Name:	Upload Attendance
Actor:	Teacher

Description:	This feature of Upload attendance shall allow the teacher to upload the students attendance taken in class.
Pre-conditions:	The user must have opened the application and must have authenticated him/her-self first.
Post-conditions:	The user would be able to upload the attendance of the students whom he/she is teaching.
Priority :	This feature is given high priority as it is one of the important features of the system.
Frequency Of Use:	The user can select this option many times so it has a high frequency of use.
Normal Course of Events:	The user selects to upload attendance The system will allow the user to upload the attendance of students who are registered to that teacher's courses.
Alternative Course:	<ol style="list-style-type: none"> 1. The user selects the option of check the attendance of students registered to his/her courses. 2. The user selects the option of Grades Check. 3. The user selects to exit the application in which case the application will close down.
Special Requirements:	
Assumptions:	
Notes and Issues:	

The user on selecting Subject Registration shall allow the user to choose the optional and compulsory subjects offered in the coming semester. In this module subjects are registered by the student users at the end of each semester for the upcoming semester.

Table 4.15: Subject Registration

Use Case ID:	04
Use Case Name:	Subject Registration
Actor:	Student
Description:	The user on selecting Subject Registration shall allow the user to choose the optional and compulsory subjects offered in the coming semester.

Pre-conditions:	The user must have opened the application and must have authenticated him/her-self first.
Post-conditions:	The users would register and enroll themselves for the subjects of next semester.
Priority :	This feature is given high priority as it is one of the important feature of the system.
Frequency Of Use:	The user selects this option once each semester so it has a medium frequency of use.
Normal Course of Events:	<ol style="list-style-type: none"> 1. The user selects to register upcoming subjects. 2. The system will display the offered subjects for the next semester and user may register him/her-self for the subjects.
Alternative Course:	<ol style="list-style-type: none"> 1. The user selects the option of Attendance Check 2. The user selects the option of Grades Check 3. The user selects the option of Improvements 4. The user selects the option of Previous records 5. The user selects the option of Balanced 6. The user selects the option of Internships/Scholarships offered and Event checks/Updates. 7. The user selects to exit the application in which case the application will close down
Special Requirements:	
Assumptions:	
Notes and Issues:	

The user on selecting Grades Check shall allow the user to view the grades of all the previous courses. This module takes care of the courses repeated/improved. It maintains the important information and all previous record for the courses that are opted to be improved.

Table 4.16: Grades check

Use Case ID:	05
Use Case Name:	Grades Check
Actor:	Student, Teacher
Description:	The user on selecting Grades Check shall allow the user to view the

	grades of all the previous courses.
Pre-conditions:	The user must have opened the application and must have authenticated him/her-self first.
Post-conditions:	The users would be able to view the list of all the previous courses taken along with the grades scored in each course.
Priority :	This feature is given low priority..
Frequency Of Use:	The user can select this feature as many times as he/she wishes but usually it's not viewed many times so it has medium frequency of use.
Normal Course of Events:	<ol style="list-style-type: none"> 1. The user selects to check grades in previous courses. 2. The system will display the list of previous courses along with the scored grades,
Alternative Course:	<p>If the user is Teacher :</p> <ol style="list-style-type: none"> 1. The user selects the option of upload attendance. 2. The user selects the option of Attendance Check 3. The user selects the option of Event checks/Updates 4. The user selects to exit the application in which case the application will close down <p>If the user is Student :</p> <ol style="list-style-type: none"> 1. The user selects the option of Attendance Check 2. The user selects the option of Improvements 3. The user selects the option of Previous records 4. The user selects the option of Balanced 5. The user selects the option of Internships/Scholarships offered and Event checks/Updates. 6. The user selects to exit the application in which case the application will close down
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	06
Use Case Name:	Improvements
Actor:	Student
Description:	The user on selecting Improvement feature shall allow the user to view pre-requisite of the course to be improved, the previous grade of that course, the time when subject is going to be offered and the registration deadlines for that course. Once the course is registered by the user, the system automatically sends an email of registration of the teacher of that concerned course.
Pre-conditions:	The user must have opened the application and must have authenticated him/her-self first.
Post-conditions:	The system would display the pre-requisites of the subject, previous records on that subject, the time when the subject will be offered and the registration deadlines to the user.
Priority :	This feature is given a medium priority
Frequency Of Use:	Frequency of use would be low as the user selects this feature only when the subjects are offered. i.e. Summer, Fall and Spring.
Normal Course of Events:	<ol style="list-style-type: none"> 1. The user selects the improvement option and course. 2. The system displays the pre-requisites of that course. 3. The system displays the previous grade in that course 4. The system displays time when subjects offered 5. The system displays the registration deadlines. 6. The system e-mails the concerned teacher of registration.
Alternative Course:	<ol style="list-style-type: none"> 1. The user may select the option of Attendance Check 2. The user may select the option of Grades Check 3. The user may select the option of Previous records 4. The user may select the option of Balanced 5. The user may select the option of Internships/Scholarships offered and Event

	checks/Updates. 6. The user may select to exit the application in which case the application will close down
Special Requirements:	
Assumptions:	
Notes and Issues:	

On selecting **Previous Record Check and Balanced credit hours** feature, system shall allow the user to view the record of the previous courses. i.e. the courses studied, the grades scored on studied courses, the credit hours completed, the remaining courses and credit hours. All these records are maintained in a database. **Previous Record** module stores all the information of the courses completed, grades scored in those courses, any improvement made in those courses, their improved grades and any extra subject completed. **Balanced Credit Hours** module stores all the information regarding the credit hours completed and the credit hours left.

Table 4.17: Previous Record Check and Balanced credit hours

Use Case ID:	07
Use Case Name:	Previous Record Check and Balanced credit hours
Actor:	Student
Description:	On selecting Previous Record Check and Balanced credit hours feature, system shall allow the user to view the record of the previous courses. i.e. the courses studied, the grades scored on studied courses, the credit hours completed, the remaining courses and credit hours. All these records are maintained in a database.
Pre-conditions:	The user must have opened the application and must have authenticated him/her-self first.
Post-conditions:	The system would display the user with list of all the courses that student has studied, their credit hours, the total credit hours studied,

	the remaining credit hours and courses. Also, the system will generate a list of the grades of previous courses along with the course names.
Priority :	This feature is given a medium priority
Frequency Of Use:	Frequency of use would be low.
Normal Course of Events:	<ol style="list-style-type: none"> 1. The user selects the Previous Record Check and Balanced credit hours option. 2. The system displays the courses completed. 3. The system displays the grade in that course. 4. The system displays the credit hours completed. 5. The system displays the credit hours left. 6. The system displays the courses left.
Alternative Course:	<ul style="list-style-type: none"> - The user may select the option of Attendance Check. - The user may select the option of Grades Check - The user may select the option of Improvements. - The user may select the option of Internships/Scholarships offered and Event checks/Updates. - The user may select to exit the application in which case the application will close down
Special Requirements:	
Assumptions:	
Notes and Issues:	

On selecting **Internship and Scholarship Check/Updates** feature, the system shall allow the user to view the latest internships, Scholarships offered at the campus along with their criteria and policies. All these records are maintained in the database of the system. **Internship check** module contains important updates regarding Monthly internships for the students of NUST. **Scholarship check** module stores important information regarding different scholarships offered at NUST and their criteria.

Table 4.18: Internship and Scholarship Check/Updates

Use Case ID:	08
Use Case Name:	Internship and Scholarship Check/Updates
Actor:	Student
Description:	On selecting Internship and Scholarship Check/Updates feature, the system shall allow the user to view the latest internships, Scholarships offered at the campus along with their criteria and policies. All these records are maintained in the database of the system.
Pre-conditions:	The user must have opened the application.
Post-conditions:	The system would result in displaying all the latest news on the internships and scholarships offered, their application forms in downloadable format and the minimum requirements needed.
Priority :	This feature is given a high priority as it needs constant updating in the database.
Frequency Of Use:	Frequency of use would be high as most student try to avail the scholarships opportunities.
Normal Course of Events:	<ol style="list-style-type: none"> 1. The user selects the Internship and Scholarship Check/Updates option. 2. The system displays the internships available. 3. The system displays a downloadable application form for internship where needed or provides with the E-mail ID to apply for internship. 4. The system displays the minimum requirements required for internship. 5. The system displays the scholarships offered. 6. The system displays the minimum criteria to avail the scholarship.
Alternative Course:	<ol style="list-style-type: none"> 1. The user may select the option of Attendance Check 2. The user may select for option of course registration. 3. The user may select the option of Improvements

	<ol style="list-style-type: none"> 4. The user may select the option of Previous records 5. The user may select the option of Balanced. 6. The user may select the option of Event Check/Updates. 7. The user may select to exit the application in which case the application will close down.
Special Requirements:	
Assumptions:	
Notes and Issues:	

On selecting **Event Check/Update** feature, the system shall allow the user to view the upcoming events at University and the calendar of events. It shall also allow the user to create a particular event. This module contains important updates regarding the events that are going to take place at NUST and is constantly updated.

Table 4.19: Event Check/Update

Use Case ID:	09
Use Case Name:	Event Check/Update
Actor:	Student
Description:	On selecting Event Check/Update feature, the system shall allow the user to view the upcoming events at University and the calendar of events. It shall also allow the user to create a particular event.
Pre-conditions:	The user must have opened the application.
Post-conditions:	The system would result in displaying all the latest events.
Priority :	This feature is given a medium priority.
Frequency Of Use:	Users may access this feature without any authentication so the frequency of use would be high.
Normal Course of Events:	<ol style="list-style-type: none"> 1. The user selects the Event Check/Update option. 2. The system lists all the upcoming events at all campuses of the university. 3. The system allows the user to create an event.
Alternative Course:	<ol style="list-style-type: none"> 1. The user may select to exit the application in which case the application will close down.

Special Requirements:	
Assumptions:	
Notes and Issues:	

4.7.5.3 Use case Diagrams of Modules

Use case diagrams for the system has been designed to give the reader of this document an overview of the system and how it is going to work under taken assumptions and perfect conditions. The use case for each module has been designed. The use cases include the Login, check and update, grade, subject registration, improvements, balanced and previous record, internship and scholarship and events.

4.7.5.3.1 Login

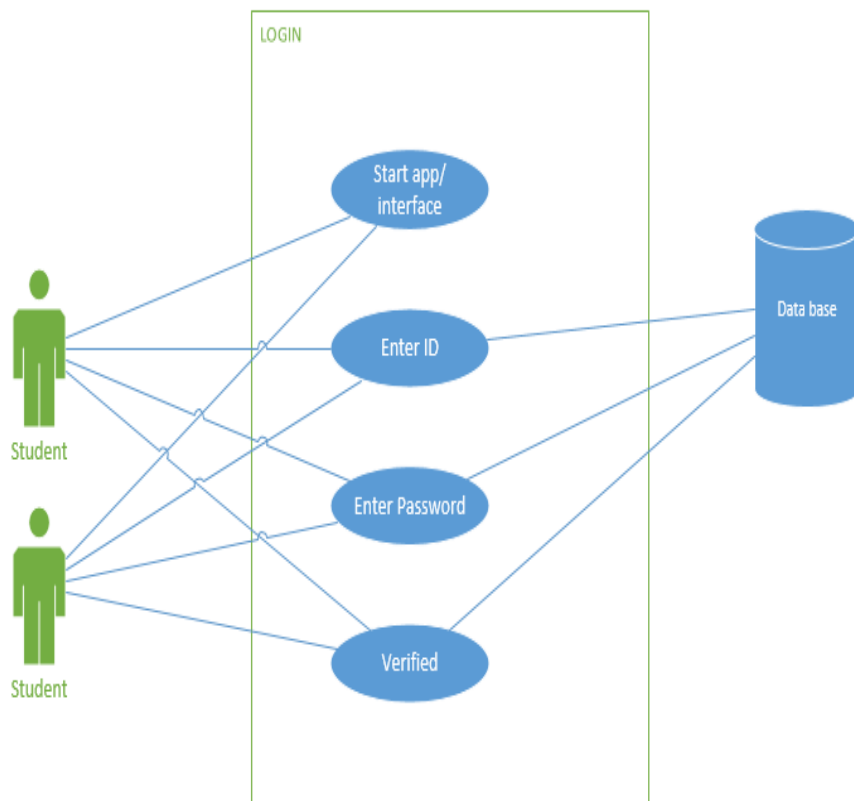


Figure 4.14: Login

4.7.5.3.2 Check and Update Attendance

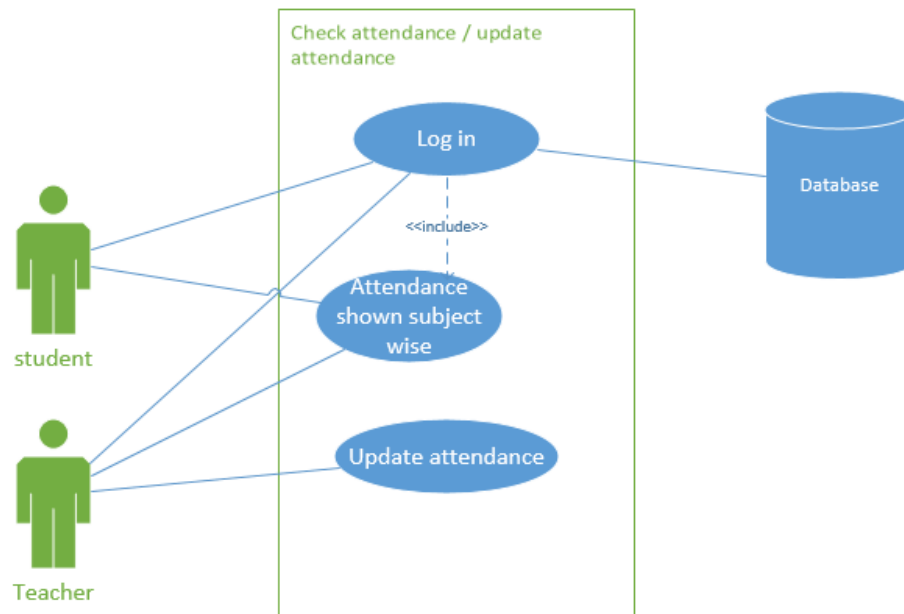


Figure 4.15: Attendance Use Case

This module has two parts, uploading attendance and checking attendance. Student's attendance record is maintained by this subsystem.

4.7.5.3.3 Grades

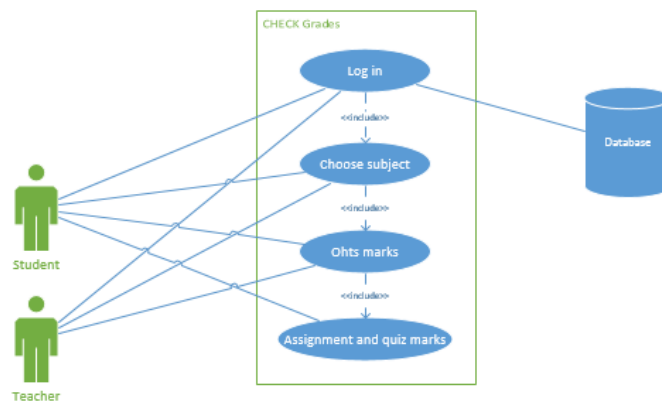


Figure 4.16: Grades Use Case

This module has all the previous data on each subject enrolled by each student of the university. All the grades scored on courses studied in the previous semesters are stored through this module.

4.7.5.3.4 Subject Registration

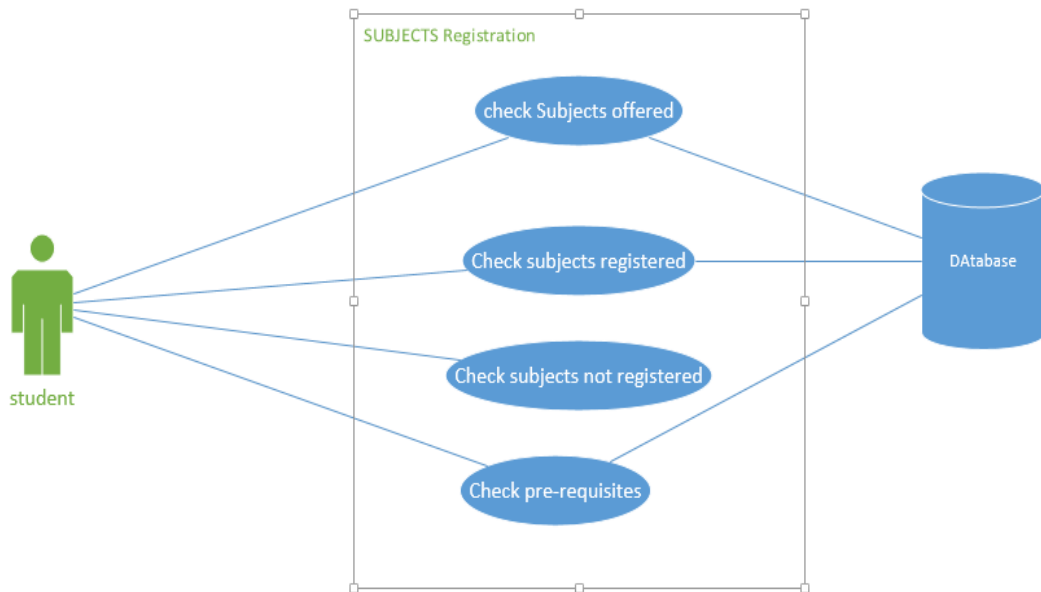


Figure 4.17: Subject Registration Use Case

In this module subjects are registered by the student users at the end of each semester for the upcoming semester.

4.7.5.3.5 Improvements

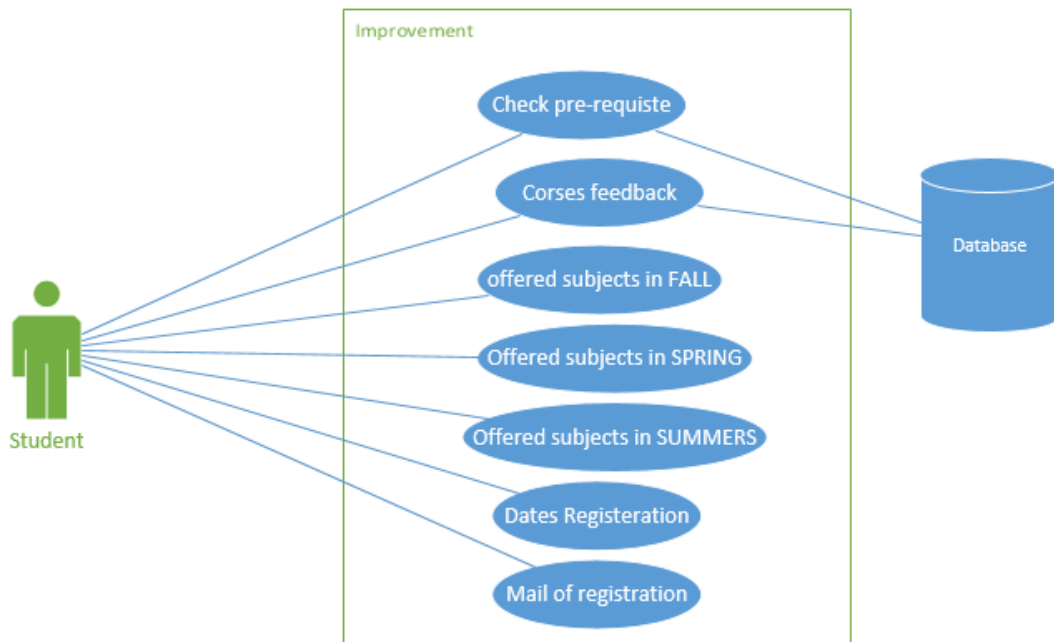


Figure 4.18: Improvements Use Case

This module takes care of the courses repeated/improved. It maintains the important information and all previous record for the courses that are opted to be improved.

4.7.5.3.6 Balanced and Previous Records

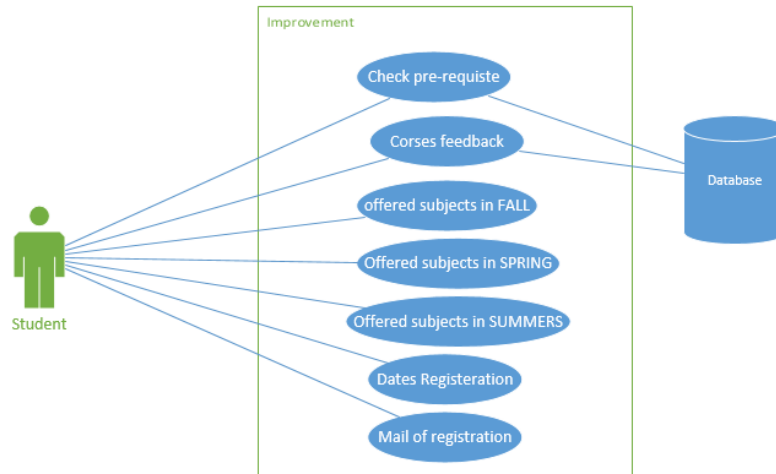


Figure 4.19: Balanced and Previous Record Use Case

Previous Record module stores all the information of the courses completed, grades scored in those courses, any improvement made in those courses, their improved grades and any extra subject completed. **Balanced** module stores all the information regarding the credit hours completed and the credit hours left.

4.7.5.3.7 Internship and Scholarships

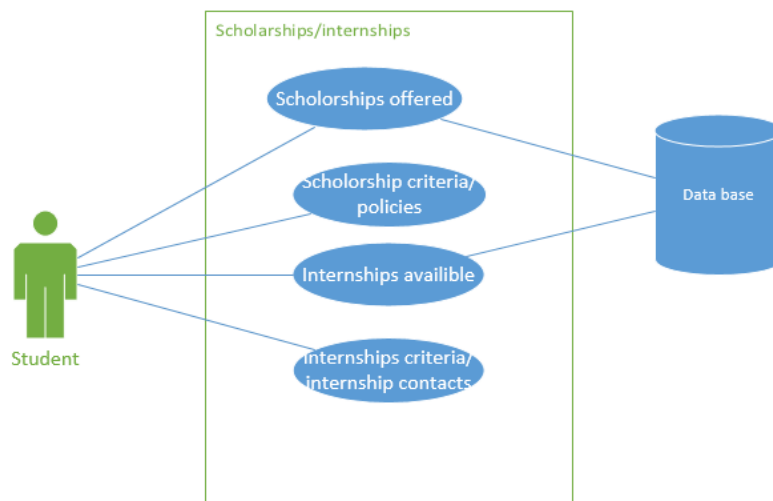


Figure 4.20: Internships/Scholarships Use Case

This module contains important updates regarding Monthly internships for the students of NUST.

4.7.5.3.8 Events

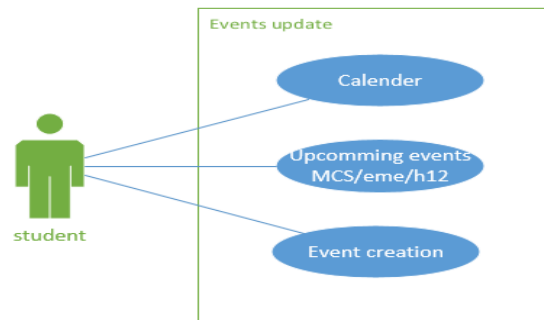


Figure 4.21: Events Use Case

This module contains important updates regarding the events that are going to take place at NUST and is constantly updated.

4.8 Conclusion

This goal of describing the software functionality based on the design of the product using different design approaches before implementing it was finally understood. Also the documents' intention to be used by the members of the project team that will implement and verify the correct functioning of the system was implemented.

5. Project Analysis and Evaluation

5.1 Introduction

In this part, we developed some test cases to test the system. To ensure the quality of the system, different test cases were produced with different testing levels for each module.

The next part of this chapter will describe the testing, testing levels and finally the unit testing for each module.

5.2 Testing

To ensure quality of the product, testing is conducted. Accuracy and efficiency of tasks performed by our system had to be tested to analyze the system and verify and validate it. Software testing techniques and results obtained are discussed in the coming sections.

5.3 Testing Levels

Separate modules were developed to provide different functionalities of the system. All of these modules were tested at different levels during development and after integration. Different levels of testing and results have been described here:

5.4 Unit Testing

Each module was designed, developed and tested individually. Each functionality was also tested separately. Detailed procedure of each test along with the expected and received results is presented below:

Test case 1 shows the creation of the profile of the user. Taking certain assumptions, data, steps and expected results into account, the results were gathered.

Table 5.1: Profile Creation

Test Case ID	1
Unit to Test	Profile Creation
Assumptions	System is up and running
Test Data	Profile details like username, course
Steps to be Executed	<ol style="list-style-type: none"> 2. User clicks the profile creation button. 3. A screen asks the user to enter user details like username, course, ,subjects, semester etc.
Expected Result	Profile is created and message is displayed
Actual Result	As Expected
Pass/Fail	Pass

Test case 2 shows the Login module of the user. Taking all the assumptions, data, steps and expected results into account, the results were gathered.

Table 5: Login

Test Case ID	2
Unit to Test	Login
Assumptions	User profile is already created
Test Data	Username, password
Steps to be Executed	<ol style="list-style-type: none"> 4. User enters username and password 5. If the password is correct, user is logged in

	6. If the password is wrong, password is asked again
Expected Result	User is logged in
Actual Result	As Expected
Pass/Fail	Pass

Test case 3 shows the updating of the profile of the user. Taking certain assumptions, data, steps and expected results into account, the results were gathered.

Table 5.3: Profile Update

Test Case ID	3
Unit to Test	Profile update
Assumptions	A user is logged in
Test Data	Course, subjects, semester
Steps to be Executed	7. User updates his information fields like course, subjects , semester 8. On saving the updates, profile is updated and success message is displayed.
Expected Result	User profile is updated.
Actual Result	As Expected
Pass/Fail	Pass

Test case 4 shows the Subject Registration of the user. Taking certain assumptions, data, steps and expected results into account, the results were gathered.

Table 5.4: Subject Registration (Offer subjects)

Test Case ID	4
Unit to Test	Subject Registration(Offer subjects)
Assumptions	9. Teacher user is logged in
Test Data	10. Subjects SPM, OS, DM etc.
Steps to be Executed	11. Teacher clicks on offer subjects 12. Teacher enters Subject details like name, course, credit hours 13. Teacher clicks save and subject is added
Expected Result	1. Subject is displayed as offered and can be applied by students to enroll in it 2. Subject resource page is created
Actual Result	As Expected
Pass/Fail	Pass

Test case 5 shows the subject registration (Apply for enrollment in subjects) of the user. Taking certain assumptions, data, steps and expected results into account, the results were gathered.

Table 5.5: Subject Registration (Apply for enrollment in subjects)

Test Case ID	5
--------------	---

Unit to Test	Subject Registration(Apply for enrollment in subjects)
Assumptions	14. A student user is logged in 15. Subjects are already offered
Test Data	Offered subjects SPM, OS, DM etc.
Steps to be Executed	1. A student selects the subjects to enroll into 2. On clicking save, enrollment is applied in the subjects
Expected Result	Student enrollment is applied in the selected subjects
Actual Result	As Expected
Pass/Fail	Pass

Test case 6 shows the subject registration (Accept Enrollment) of the user. Taking certain assumptions, data, steps and expected results into account, the results were gathered

Table 5.6: Subject Registration (Accept Enrollment)

Test Case ID	6
Unit to Test	Subject Registration(Accept Enrollment)
Assumptions	1. A teacher user is logged in 2. Subjects are already offered 3. Students enrollment requests are already applied
Test Data	Student enrollment requests
Steps to be Executed	Teacher accepts the student's enrollment requests.

Expected Result	<ol style="list-style-type: none"> 1. Students are enrolled 2. Students can view the subject's resource page
Actual Result	As Expected
Pass/Fail	Pass

Test case 7 shows the Attendance of the user. Taking certain assumptions, data, steps and expected results into account, the results were gathered

Table 5.7: Attendance(post/upload attendance)

Test Case ID	7
Unit to Test	Attendance(post/upload attendance)
Assumptions	<ol style="list-style-type: none"> 1. A teacher user is logged in 2. Students attendance record is available
Test Data	Students attendance record
Steps to be Executed	<ol style="list-style-type: none"> 1. Teacher marks the student's attendance as absent or present 2. On clicking save, students attendance is uploaded and percentage is calculated
Expected Result	Student's attendance is marked and percentage is calculated
Actual Result	As Expected
Pass/Fail	Pass

Test case 8 shows the Attendance (view attendance) of the user. Taking certain assumptions, data, steps and expected results into account, the results were gathered

Table 5.8: Attendance (view attendance)

Test Case ID	8
Unit to Test	Attendance(view attendance)
Assumptions	<ol style="list-style-type: none"> 1. A student user is logged in 2. He is already registered in the course
Test Data	Student's attendance record
Steps to be Executed	<ol style="list-style-type: none"> 1. Student selects the subject whose attendance is to be viewed
Expected Result	Student's attendance and percentage is visible
Actual Result	As Expected
Pass/Fail	Pass

Test case 9 shows the Attendance (update/edit attendance) of the user. Taking certain assumptions, data, steps and expected results into account, the results were gathered

Table 5.9: Attendance (update/edit attendance)

Test Case ID	9
Unit to Test	Attendance(update/edit attendance)
Assumptions	<ol style="list-style-type: none"> 1. A teacher user is logged in 2. Student's attendance record is already available
Test Data	Student's attendance record
Steps to be Executed	<ol style="list-style-type: none"> 1. Teacher selects the user attendance record

	to be edited/updated 2. Teacher edits/updates students attendance record 3. New attendance is calculated.
Expected Result	New attendance is marked and percentage is calculated
Actual Result	As Expected
Pass/Fail	Pass

Test case 10 shows the Grades (post/update) of the user. Taking certain assumptions, data, steps and expected results into account, the results were gathered

Table 5.10: Grades (post/update)

Test Case ID	10
Unit to Test	Grades(post/update)
Assumptions	1. A teacher user is logged in 2. Student has taken the exams and completed the semester
Test Data	Student's Grades
Steps to be Executed	1. Teacher enters marks earned by students in exams 2. The student grade is calculated based on marks obtained in semester exams 3. Teacher posts the grade
Expected Result	Grade is calculated and posted
Actual Result	As Expected

Pass/Fail	Pass
-----------	------

Test case 11 shows the Grades (view) of the user. Taking certain assumptions, data, steps and expected results into account, the results were gathered

Table 5.11: Grades (view)

Test Case ID	11
Unit to Test	Grades (View)
Assumptions	A student user is logged in.
Test Data	Student's grades
Steps to be Executed	<ol style="list-style-type: none"> 1. Student selects the subject to view the grade 2. Students achieved grade is visible
Expected Result	Student's results and grades are viewed successfully
Actual Result	As Expected
Pass/Fail	Pass

Test case 12 shows the previous record of the user. Taking certain assumptions, data, steps and expected results into account, the results were gathered

Table 5.12: Previous record

Test Case ID	12
--------------	----

Unit to Test	Previous record
Assumptions	1. Student's previous record is available 2. Student has studied some courses and scored some grades
Test Data	-
Steps to be Executed	1. Student selects the previous record to be viewed 2. Courses studied, grades scored, credit hours completed and remaining credit hours are visible
Expected Result	Previous record is maintained and is visible
Actual Result	As Expected
Pass/Fail	Pass

Test case 13 shows the Events (post and check events) of the user. Taking certain assumptions, data, steps and expected results into account, the results were gathered

Table 5.13: Events (post and check events)

Test Case ID	13
Unit to Test	Events(post and check events)
Assumptions	1. A user is logged in
Test Data	A student academic event (workshop, scholarship)
Steps to be Executed	1. The user selects the event creation tab

	2. User enters the event details 3. On save, the event is created and is visible to other users
Expected Result	Event is created and is visible by other users.
Actual Result	As Expected
Pass/Fail	Pass

6. Conclusion and Future Work

After the completion of NUSTBOOK, there would be a lot of ease to the student of NUST, in managing their courses, assignments, attendances and a lot more that they were not able to do with the previous system.

After NUSTBOOK, there is no need to develop another LMS system for the students of NUST considering the present requirements. Although with the change in the requirements, another version of NUSTBOOK can be developed to meet those requirements.

Appendix A: User Manual

Table of contents

1. General Information	1
1.1 System Overview	1
1.2 Project References	1
1.3 Organization of the Manual	1
1.4 Acronyms and abbreviations.....	2
2. System Summary	3
2.1 System Configuration	3
2.2 User Access Levels	3
2.3 Contingencies and Alternate mode of Operations	3
3. Using the System	4
3.1 Logging in.....	4
3.2 Successful Login.....	4

7. General Information

General Information section explains the system and the purpose for which it is developed.

7.1 System Overview

NUST Book is an application that will aid in accessing the LMS. This application shall guide the student's through-out their four-year degree program. The application shall keep track of the students' progress and attendance of the current semester as well as the previous semesters. The application shall provide guidelines to the students on how to improve their results. The application will use web scraping, html parsing to create an intuitive interface for exploring the LMS. The application will access LMS to fetch data required for the application to operate properly and therefore the application requires the use of internet to be fully functional

7.2 Project References

The only reference used to make this project was the previously implemented NUST-LMS.

- <https://lms.nust.edu.pk>

7.3 Organization of the Manual

The User Manual is divided in three sections.

- General Information
- System Summary
- Interfaces and Functionality

General Information section explains the system and the purpose for which it is developed.

NUST BOOK

System Summary section gives you information about the system configuration components and different modules of the system.

Interface and Functionality introduce you to the system interface and its functionality.

7.4 Acronyms and abbreviations

App: Application

LMS: Learning Management System

SRS: Software Requirement Specification

NUST: National University of Science & Technology

MCS: Military College of Signals

HTTPS: Hyper Text Transfer Protocol Secure

OS: Operating System

OHT's: One Hour Tests

UI: User Interface

App: Application

8. System Summary

System summary provides the general overview of the system. The summary outlines the uses of the system's software requirement, system's configuration, user access levels and system's behavior in case of any contingencies.

8.1 System Configuration

NUSTBOOK operates on Windows based OS.

8.1.1 System Components

NUSTBOOK is a software system and does not require any hardware components.

8.2 User Access Levels

- Administrator has full access over the system. Admin user can change login information
- Student user can check the updates on everything.
- Faculty user can upload and check updates on everthing.

8.3 Contingencies and Alternate mode of Operations

The system shall be available at all time if there is a permanent connection to the internet as it depends on the connectivity to the internet.

9. Using the System

9.1 Logging in

To prevent the unauthorized use of the system the user will insert its user name and password to log in to the system.



Figure 9.1: Login Screen

Step1: Enter your specific user name that has been provided with you along with the other reference material

Step 2: Enter your specific password that has been provided with you along with the other reference material

Step 3: Click the Login Button

9.2 Successful Login

A screen will display as follows which will indicate that the system is now ready to start and all manual controls will now disengage.

Appendix B: References

1. We learned PHP basics from different sources, one of the sources is code academy.
<https://codeacademy.org>
2. We had a great help from our faculty members of NUST.
3. The previously implemented NUST-LMS was a great help in designing this system.
<https://lms.nust.edu.pk>