Group Project group E technical report

by Sagun Rupakheti

Submission date: 19-Jul-2020 04:39PM (UTC+0100)
Submission ID: 131708502
File name: group-E-technical-report.docx (10.09M)
Word count: 17118
Character count: 91219

Course: BSc Computing (SE)	Year: 2	CSY2027

Group Project	Title: The Design and Development of a Course Management Software System

Date due out:	Date due in:	Extension date:	Extension agreed by:

Student Names (List each member of the group)	
Aditya Jha- 19436752 **Anish Gelal- 19436754** **Grishma Duwadi- 19436758** **Manish Thapa- 19436755** **Prekshya Neupane- 19436779** **Sagun Rupakheti - 19436773** **Sushank Aryal- 19436759**	

Student comment, specific request for feedback etc.	Marker's General View of the work

ASSESSMENT FEEDBACK:

RATING SCALE	Excellent	Good	Satisfactory	Needs some more work	Needs much more work
Elicitation Plan/ Interview(s)/Findings (10%)					
Requirement Specifications Documentation (10%)					
System Design Documentation (25%)					
Prototype Functionality and Quality of Application Code (35%)					
Test/Evaluation Strategy (10%)					

Group Cohesion, Teamwork and Project Management/ System Presentation (10%)					

Specific aspects of your assignment that the marker likes:	Specific aspects of your assignment that need more work:

Tutor's Signature:	Date:	Grade:

Table of Contents

Abstract 15

1. Introduction 17

1.1 Project Background 17

1.2. Project aims and objective 18

1.3. Project development methodology 19

2. Requirements Engineering 19

2.1. Elicitation Activities 20

2.1.1. Interview Plans 20

2.1.1. Interview Findings 25

2.1.3. Other problem domain research 37

2.1.3.1. Comparable Software System Review 37

2.1.3.1.1. Moodle 37

2.1.3.1.2. Blackboard 40

2.1.3.1.3. NILE 42

2.1.3.2 Development Relevant Legislation: - 43

2.1.3.2.1. Data Protection Act 43

2.1.3.2.2. Digital Economy Act 44

2.1.3.2.3. Computer Misuse Act 44

2.1.3.3 Any other relevant problem domain investigation data 44

2.2. Requirements Specification 45

2.2.1. Problem Domain Description 45

2.2.1.1. Existing Business Operation: - 45

2.2.1.1.1. UCAS and letter 46

2.2.1.1.2. Id generation and student folder 47

2.2.1.1.3. Welcome Week 48

2.2.1.2. Summary of existing business limitations 51

2.2.2. Functional Requirements 51

2.2.2.1. Required Features of the System 51

2.2.2.1.1. Database and backup 52

2.2.2.1.2. Students records 52

2.2.2.1.3. Staff records 53

2.2.2.1.4. Course Records 53

2.2.2.1.5. Module Management 54

2.2.2.1.6. Assignment Management 54

2.2.2.1.7. Attendance Management 54

2.2.2.1.8. Personal Tutor Management 54

2.2.2.1.9. Grades Management 54

2.2.2.1.10. Module Content Management 55

2.2.2.1.11. Announcement Management 55

2.2.2.2. Access Rights/ Level of Access 55

2.2.2.2.1. Administrator, Staff and Student Management 56

2.2.2.2.2. Course Management 60

2.2.2.2.3. Module and Content Management 61

2.2.2.2.4. Assignment Management 64

2.2.2.2.5. Grade Management 66

2.2.2.2.6. Attendance Management 68

2.2.2.2.7. Personal Tutor Management 69

2.2.2.3. Output formats for onscreen displays 70

2.2.2.3.1. Enrolled to the system 71

2.2.2.3.2. Low Attendance 72

2.2.2.3.3. Grades Published 73

2.2.2.3.4. Assignment Published 74

2.2.2.3.5. Assignment Submitted 75

2.2.2.3.6. PAT meeting schedule 76

2.2.3. Performance Requirements 77

2.2.3.1. Speed 77

2.2.3.2. Capacity 77

2.2.3.3. Reliability 78

2.2.3.4. Usability 78

2.2.4. Design Constraints 78

2.2.5. Commercial Constraints 80

3. System Analysis & Design 82

3.1. Preliminary Design Stages 82

3.1.1. Textual analysis 83

3.1.2. Significant Event Analysis 84

3.1.3 Commands Queries and Constraints 92

3.2 Detailed Static System Designs 95

3.2.1. First Draft BON System Architecture Diagram 95

3.2.2. Use Case Diagrams 98

3.2.2.1. Administrator Use Case Diagram: 98

3.2.2.2. Module Leader Use Case Diagram 100

3.2.2.3. Student Use Case Diagram 101

3.2.3. Use case documentations 102

3.2.3.1. UC1 Administrator tab 102

3.2.3.2. UC2 Module Leader tab 103

3.2.3.3. UC3 Announcements tab 104

3.2.3.4. UC4 Courses tab 104

3.2.3.5. UC5 Students tab 105

3.2.4. Class diagram 106

3.3 Detailed Dynamic System Designs 107

3.3.1. System Sequence Diagrams 107

3.3.2. System Collaboration/ Communication Diagrams 110

3.4. System Database Design 111

3.4.1 E-R Model 112

3.4.2 Attribute Listings 113

4. System Interface Designs 115

4.1. Draft Interface Designs 115

4.1.1. Wireframes 115

4.1.2. System Navigation Diagram 118

4.1.3. System Screen mock-ups 120

4.1.4. System Activity Event Diagrams 121

5. System Build and Technical Notes 128

5.1 Technically Difficult Code Sections 128

5.2 On-going Testing Methodology 131

5.3 Final System Interface Displays 131

6. Test Strategy 137

6.1 Overview of Test Strategy 137

6.2 Sample Test Results 138

6.2.1. Administrator Testing 138

6.2.2. Module Leaders Testing 157

6.2.3. Students Testing 167

6.2.4. All user groups 169

7. System Evaluation 171

7.1 Usability Evaluation Strategy 171

7.2 System Pilot Trials 171

7.3 System Trial Results 178

8. Project Conclusions 179

9. Project Management 181

9.1. Project Gantt Chart/WBDS/Activity List 182

9.2. Project Meeting Minutes 184

9.3. Project Quality Plan/ Strategy 195

9.4. Project Work Log 198

10. References 207

11. Appendix 208

Table of Figures:

Figure 1. 1 – Moodle website 38

Figure 1. 2 Blackboard website 40

Figure 1. 3 Flowchart showcasing the ongoing process of admission through UCAS 46

Figure 1. 4 Flowchart showcasing the generation of Id and storage 48

Figure 1. 5 Flowchart showcasing the welcome week and admission criteria 50

Figure 1. 6 BON System Architecture Diagram 96

Figure 1. 7 BON System Architecture Diagram Flow 97

Figure 1. 8 Admin Use Case Diagram 99

Figure 1. 9 Module Leader Use Case Diagram 100

Figure 1. 10 Student Use Case Diagram 101

Figure 1. 11 System Class Diagram 106

Figure 1. 12 Sequence diagram for adding an administrator 108

Figure 1. 13 Sequence diagram for adding a module leader 108

Figure 1. 14. Sequence diagram for adding a student to the system 109

Figure 1. 15 Collaboration diagram for adding an administrator 110

Figure 1. 16 Collaboration diagram for adding a module leader 110

Figure 1. 17. Collaboration diagram for adding a student to the system 111

Figure 1. 18. Wireframe showing students list 116

Figure 1. 19 Wireframe showing student's details 117

Figure 1. 24. Detailed navigation of module leader 118

Figure 1. 25 Full system navigation diagram 119

Figure 1. 26. Admin Dashboard 120

Figure 1. 27 The admin clicks on Students tab 121

Figure 1. 36. Flowchart showing the process of adding the course 121

Figure 1. 37. Flowchart showing the process of adding a module leader 122

Figure 1. 38 Flowchart showing the process of adding a module 123

Figure 1. 39. Flowchart showing the admission process 124

Figure 1. 40. Flowchart showing the process of modifying the module leader 125

Figure 1. 41. Flowchart showing the process of modifying the course 126

Figure 1. 42. Flowchart showing the process of adding announcements 127

Figure 1. 43Admin Dashboard 132

Figure 1. 44Admission process 132

Figure 1. 45Module leader dashboard 133

Figure 1. 46Module view 134

Figure 1. 47 Attendance view 134

Figure 1. 48 PAT communication 135

Figure 1. 49 Profile view 135

Figure 1. 50 Time-table view 136

Figure 1. 51 Announcements view 136

Figure 1. 52 Admin insertion successful and failure 153

Figure 1. 53 Module Leader insertion success and failure 154

Figure 1. 54. Course insertion success and failure 155

Figure 1. 55 Module insertion success 156

Figure 1. 56. Add Module activities success and failure 163

Figure 1. 57. Assignment upload success and failure 164

Figure 1. 58. Adding attendance 165

Figure 1. 59. Low-attendance warning 166

Figure 1. 60 Wrong login credentials 170

Figure 1. 61 Gantt chart to showcase the time elapsed in the documentation of the project 182

..... Figure 1. 62. . Gantt chart to showcase the time elapsed in the code and functionalities implementation of the project 183

Figure 1. 20. Wireframe showing edit course page 210

Figure 1. 21. Wireframe showing list of module/courses 211

Figure 1. 22. Wireframe showing students records to edit 212

Figure 1. 23. Wireframe showing admission process 213

Figure 1. 28. The admin chooses to upload a file 218

Figure 1. 29 The admin selects a UCAS file to upload 218

Figure 1. 30 The UCAS file is uploaded..... 219

Figure 1. 31. The UCAS file is uploaded..... 220

Figure 1. 32. Student dashboard 220

Figure 1. 33. Student views module materials..... 220

Figure 1. 34. Student views module assignments 221

Figure 1. 35. Student uploads assignment 221

Abstract

Woodlands University College decided to switch from clerical based system to a fully digitalized course management system. The college had not undergone any digitalization so far. A series of interviews were taken with different designated people to have a better understanding of client requirements. We had interviews with Mr Adam Blake, Mr. Mark Williams, Mr, Simon White and Mr. Raj Singh.

The functionalities were detected, and the project planning was started. After the interview process, the UI design along with other documentation activities like event diagrams, functionalities, classes identifications were commenced. The functionalities were sorted based on their priority orders. The priority highest prioritized functionalities took longer to work on whereas the lower ones were quicker. Activity event diagrams and navigation diagrams played a vital role in shaping the functionalities and flow of the system. Alongside the functional requirements, the requirements engineering portion went further with comparing other existing systems and building our system based on their weaknesses and strength. The main inspiration for the system was taken to be NILE and Moodle. Dynamic and static designs were made to further enhance the system knowledge and then the implementation portion was started.

For the implementation, PHP framework CodeIgniter was chosen and for the database, MYSQL was used. Alongside the building of the system, testing was done extensively. Black box testing methods were used to test all the possible test cases and check their expected outcomes with the actual outcomes. Project Management has been given a major priority to detect the time elapsed and to showcase that the high priority functionalities were directly proportional to time consumed.

Overall, the project development process has been a major learning curve for all the members which also gave a small insight into the work that is put together to build a real-time working system.

1. Introduction

In this digitalized time most of the colleges and education forms are using different computerized course management system in order to manage all the courses of the students for the ease and benefits of quality education.

Woodlands University College (WUC) is a small higher education institution which demands the computerized course management system for the computing degree course. Since, the woodland university college is facing problems in handling the records, files of the students and teachers of the university so it demands a software which can manage the student records, staff records, course records, module management, assignment management, attendance records, personal tutor management, timetable management, diary management and report management of computing department.

The main objective of this report is to describe the plans and development of the website for the computing degree at WUC. The project includes short description on requirement engineering, requirement specification, problem domain description, functional requirements, design constraint, project management etc. which describes the proper functionality of the system.

The project is implemented using PHP framework "CodeIgniter" which is used in developing dynamic websites with PHP and bootstrap is also used for the user interface. An object-oriented approach has been taken. MYSQL is used for database storage and usability.

1.1 Project Background

Woodlands University College (WUC) is a small institution of higher education which offers a wide range of degree courses. Dr. Simon White is the Computing Degree course leader at WUC. As Dr. White requested to develop computerized course management system for computing department to manage the data of student, staffs and current student can enroll in different courses according to their module. This software is made to solve the existing problem of the university and to make the work easier as University was using clerical system.

Some of the features of the website are:

- Student records along with the grades graded by the tutors
- Staff records can be recorded
- Courses can be uploaded by the module tutor and students are able to download the file
- Assignments of the different modules are uploaded by the module leader or course leader in the website
- Personal tutor are assigned to the student and student are able to confess their problems to the tutors
- Time table of the classes are uploaded by the tutor and are displayed to the students
- Students are able to maintain the diary

1.2. Project aims and objective

The main aims and objective of the project is to develop the computerized course management system which makes the work easier to the university for the management of records/data of the students, staff, module etc. of computing degree. The pilot system stores the records of students, staff, course. It also handles assignment management, attendance records, timetable management, diary management, report management of the students so that it would be easier for the university to maintain such records.

Since system is for computing degree, course leaders should be able to upload the study materials in the system. Hence, students should be able to download the study materials

provided by the course leaders. Assignment/exams should be conducted and grades of different modules should also be displayed if, student is graded by the module leaders. Students records should be recorded by the course leaders.
Likewise, staff records can also be maintained in the system.

1.3. Project development methodology

Four methods of development are listed below:

- Inception: It deals with the understanding of the project. Basic information were provided by the course leader, course administrator, senior lecturer and the computing students of the WUC.
- Elicitation activities: It deals with the researching the website's requirement, which was achieved through the interviews of the course leader, course administrator, senior lecturer and the computing students of the WUC.
- Design and analysis: Websites were designed as per client requirements and were programmed as per designed wireframe and mockups. Finally, the prototype was tested and were approved.
- Deployment: Now, the website was handed over to the course leader of WUC.

2. Requirements Engineering

In Requirement Engineering all the information about feature and services that system should provide has been gather. It help to understand customer desires and help to analysis system feasibility in reasonable price.

2.1. Elicitation Activities

To know more about the required system for WCU different method were used for collecting information which includes interview of the client for extracting information.

2.1.1. Interview Plans

Various information were collected and accordingly various interview question were prepared for each individual interviewee.

Here are the question that were asked to the client during interview

Problem Domain Entities	Elicitation Strategy
Dr. Simon White (The computing course leader)	1. What are the reasons to make the decision to switch from a clerical? 2. Have you tried switching the paper-based system to a computerized one? If so, what made it a failure? 3. What are the major difficulties faced due to the current system?

	4. What are your main expectations from the new system? 5. Is there any constraint or information regarding the legislation? 6. Is there any constraint or information regarding the security and privacy policies? 7. Are there any preferences for the programming language to be used? 8. Have you used any sort of course management system before? 9. Will there be user data provided for the pilot system's database? 10. Is there any specific feature that you would like to add or remove? 11. Any training session required for the system usage?

Mr. Adam Blake (Course administrator for computing program)	1. How exactly would you describe your role for the computing department? 2. Being an administrator, can you describe the different user groups and their roles including the granted access and priorities? 3. What would be the methods or criteria for addition of new user to the system? 4. How would the announcements and notices be categorized for all user groups? 5. Present way of attendance. Do you want to change the way of taking attendance? 6. Could system allow student to submit assignment? 7. Would module able to edit their time table or admin do it for all modules? 8. Would you want any search facility to be provided for easy access of information? 9. Are there any other extra features you would like to suggest?

	10. Cost of the project and deadline of the project
Dr. Raj Singh (A senior lecturer, module leader and a personal tutor)	1. What are your thoughts on the ongoing attempt to change the clerical system? 2. What are the major difficulties faced due to current system? 3. How could you like to manage module content? 4. What are you suggestion on assignment management?

	5. How would you suggest the discussion forums to be like? 6. On what instances would you want the announcements and notices to be sent to the students? 7. Should there be any specific method of communication between the PAT and the student?
Mr. Mark Williams (A student of computing course)	1. What are your thoughts on the ongoing attempt to change the clerical system? 2. As a student, what are the features you would like to see in your course management system? 3. How could you like to communicate with pat? 4. How would you want to view the course and module details along with study materials? 5. Are there any suggestions regarding the UI of the system? 6. Would a training program for the system be beneficial for all the student?

2.1.1. Interview Findings

This section is about the finding of an interview with the core member of WCU.

Interview Title: Initial Interview with Dr. Simon White

Interview Date: 2019-11-25

Duration: 30 minutes

Person in attendance:

- Sagun Rupakheti
- Grishma Duwadi
- Manish Thapa
- Aditya Jha
- Susank Aryal
- Anish Gelal
- Prekshya Neupane

Interviewer	**Question Number:**	**Question- Client Response**

Sagun Rupakheti	01	**What are the reasons to make the decision to switch from a clerical system to a computerized system?** Lots of problems were being faced due to clerical based system such as loss of data and vital documents, difficult to manage and search, personal information security is being compromised.
	02	**Have you tried switching the paper-based system to a computerized one? If so, what made it a failure?** No, we've not tried that before it's for the first time.
	03	**What are the major difficulties faced due to the current system?** The major difficulties are to take days for record management for the teachers and student and paperwork required for updating informations.
	04	**What are your main expectations from the new system?** We have a list of major expectations from the system: • We must be able to access any person through internet for e.g. email. • The system must be user friendly and easy to use. • The system must have security and proper data protection.

		• Different privileges for user where course leader and system can control overall system, module leader were responsible for managing module activity and student can view courses they are enrolled in, their pat information and personal details.
	05	**Is there any constraint or information regarding the legislation?** We are not very knowledgeable in this field which is the reason we have hired you all. We would like you to research on this and incorporate any necessary legislative acts. Some of them which I know of are: ➢ Data Protection Act ➢ accessibility for disable peoples
	06	**Is there any constraint or information regarding the security and privacy policies?** We would like all the assignment files to be private and grades should be encrypted with strong password and network part should be strong. Any user's private information must be protected.
	07	**Are there any preferences for the programming language to be used?**

		We do not have any preference, any language i.e. safe, secure and best. We would definitely like you to use a OOP approach.
	08	**Have you used any sort of course management system before?** Not for the WUC but I have used Moodle in the past which is a vast course management system.
	09	**Will there be user data provided for the pilot system's database?** No real-time data will be provided as the system is a pilot system as well as for privacy reasons. Dummy data can be used to demonstrate the functionalities.
	10	**Is there any specific feature that you would like to add or remove?** We would definitely like the archived data to be stored as we have a major problem of storage in this clerically based system. We would at least like to store the data for 10years are dormancy of any WUC member.
	11	**Any training session required for the system usage?** We would like the system to be as much user friendly as possible. If required and requested by our students we would definitely like to have a training session.

Interview Title: Initial Interview with Mr. Adam Blake

Interview Date: 2019-11-26

Duration: 42 minutes

Person in attendance:

- Sagun Rupakheti
- Grishma Duwadi
- Manish Thapa
- Aditya Jha
- Susank Aryal
- Anish Gelal
- Prekshya Neupane

Interviewer	**Question Number:**	**Question- Client Response**
Sagun Rupakheti	01	**How exactly would you describe your role for the computing department?** I am the administrator/ course leader of the computing department. My main role to be precise is management of overall system, management of admission process and other administration works.
	02	**Being an administrator, can you describe the different user groups and their roles including the granted access and priorities?** We would like the system to be friendly for 3 user groups exactly. Three group of users and they are:- ➢ The first would be the Courser leader and admins. They will have the highest privilege and will be able to add/edit/delete

		any major operations in the system which are of top-tier priority. ➢ The second user group would be academic staffs/ module leaders. They will have semi-privilege and will be able to manage and change their own module information/activity. We would also like module leaders to be able to grade the assignments. ➢ The third user group would be students. They must be able to view module information they are enrolled in and personal details.
	03	**What would be the methods or criteria for addition of new user to the system?** Firstly, student apply for UCAS and those apply form details were sent to WCU as excel file and system should able to upload that excel file and for interested student conditional letter is sent with unique 8-digit student id number. Only 10k student are only admitted and those who just applied but did not took admission information is stored for 10 years as dormant. There is one more condition where student case status is changed into dormant if they fail to submit all the admission related document otherwise status is changed into live.
	04	**How would the announcements and notices be categorized for all user groups?** There are two levels of announcements. Module level announcements which apply to the respective modules of the module leaders and Information level announcement which applies to all the users of the system.

	05	**What is the present way of attendance. Do you want to change the way of taking attendance?** Currently we have a paper-based attendance system and there are three codes for attendance such as X,0 and A where X is absent, A is authorized absent and 0 is present. We would just like to be able to update attendance in the system and if student has attendance less than 40% notification/warning email should be sent.
	06	**Could system allow student to submit assignment?** Yes, module leaders will be able to upload assignment information and exam timetable and students should able to submit assignment within given period of time. If the student submits any assignment above the deadline the student's work must be marked as late-submission.
	07	**Would module leaders be able to edit their timetable or will admin do it for all modules?** Admin will manage all the timetable for all the modules and mainly the timetable needs to be course oriented.
	08	**Would you want any search facility to be provided for easy access of information?** Yes, search facility would be a great plus point for the system. Specially for the admins who have to be dealing with a very large amount of data.

	09	**Are there any other extra features you would like to suggest?** Yes, we have a series of extra features we would like to add. Firstly, it would be great if we could assign PAT to the students either automatically or manually and the PAT to be notified for the pat appointment. We would also like the students to be able to request for change of their personal information and the admin to be able to review and accept the changes. It would be great if modules could be created and module activities, assignments can be added to the module. The assignments should work based of the deadline and the students must be able to submit the assignments. The assignment then can be graded by the respective module leader. If we could also have a diary/notes management, it would be easier to keep notes of the tasks we are supposed to be doing.
	10	**What is the cost and deadline of the project?** As this is pilot project so the cost of the project is around 20k pounds and deadline is the end of March 2020.

Interview Title: Initial Interview with Dr. Raj Singh

Interview Date: 2019-11-28

Duration: 13 minutes

Person in attendance:

- Sagun Rupakheti
- Grishma Duwadi
- Manish Thapa
- Aditya Jha
- Susank Aryal
- Anish Gelal
- Prekshya Neupane

Interviewer	**Question Number:**	**Question- Client Response**
Sagun Rupakheti	01	**What are your thoughts on the ongoing attempt to change the clerical system?** As the current system is paper based so there are many problems such as difficulties in managing records. We are very hopeful of the system to change the difficulties and make the day to day task easier.
	02	**What are the major difficulties faced due to current system?** The major difficulties from my perspective is managing the timetable, exam schedule and student record attendance. Keeping track of all of these manually is a difficult task.
	03	**How would you like to manage module content?**

		It would be helpful if we could upload the module contents and the students could access them. However, if the module leaders want, the visibility could be changed.
	04	**What are your suggestions on assignment management?** It would be great if we could add assignment content and manage the deadline. The grading system could also be implemented which would make marking easier.
	05	**How would you suggest the discussion forums to be like?** Students could have one on one discussions with their respective PAT's or module leaders. However, this feature is not mandatory.
	06	**On what instances would you want the announcements and notices to be sent to the students?** I would like the students to be notified if assignment has been posted and would also like to send warnings to the students having low attendance in my module.
	07	**Should there be any specific method of communication between the PAT and the student?** It's better to use chat room or student will directly visit to their PAT. It could also be implemented as an email forum where the PAT could select the date and time for in-person meetings.

Interview Title: Initial Interview with Mr. Mark Williams

Interview Date: 2019-11-28

Duration: 9 minutes

Person in attendance:

- Sagun Rupakheti
- Grishma Duwadi
- Manish Thapa
- Aditya Jha
- Susank Aryal
- Anish Gelal
- Prekshya Neupane

Interviewer	**Question Number:**	**Question- Client Response**
Sagun Rupakheti	01	**What are your thoughts on the ongoing attempt to change the clerical system?** We are very happy that the university is changing from a clerical system to digital system. It has been a very difficult task to submit assignments in person.
	02	**As a student, what are the features you would like to see in your course management system?** If all module material could be accessible to student, it would be helpful. It would also be better if we could submit assignments and view our grades .

	03	**How could you like to communicate with PAT?** This system should able to make appointments with the PAT teachers which would schedule meetings.
	04	**How would you want to view the course and module details along with study materials?** Related module link should be available if that is visible to student.
	05	**Are there any suggestions regarding the UI of the system?** It should be user-friendly and attractive. The UI must also be easy to use for us students.
	06	**Would a training program for the system be beneficial for all the student?** Yes, if proper training is given then students can use it in more systematically and properly so training could be beneficial.

2.1.3. Other problem domain research

2.1.3.1. Comparable Software System Review

To know about more about required and to understand the usability of the system we had to compare similar type of per-existing system and some of them are mentioned below.

2.1.3.1.1. Moodle

Moodle is open source online based course management system. Moodle is being used by many university all around the globe. The positive accept and negative accept is discussed below.

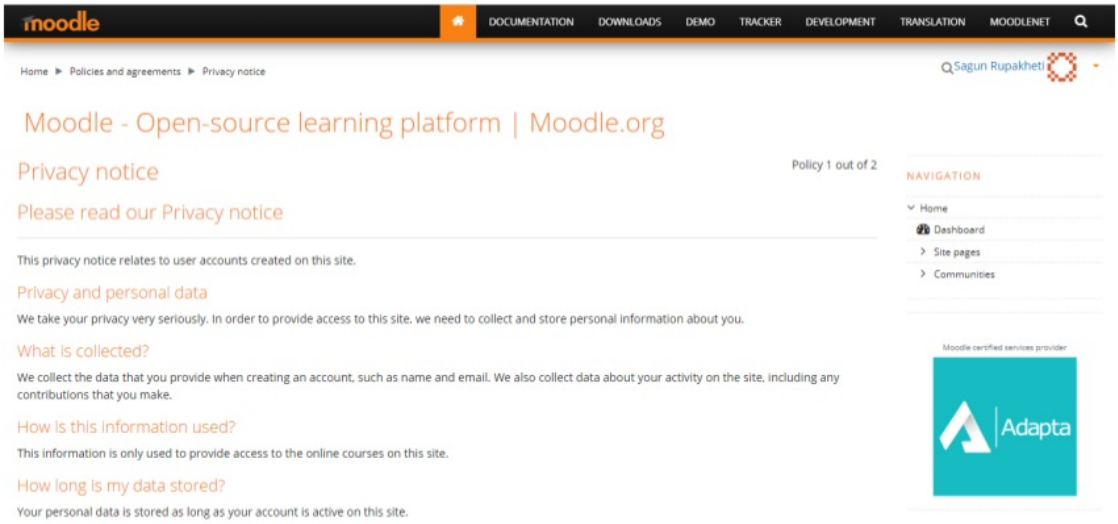


Figure 1. 1 – Moodle website

Positive aspects	Negative aspects
• Moodle is has easy navigation and compatible for all kinds of devices • Moodle can keep track of academic date such as course dateline, academic meetings. • Moodle can be customized as per the users requirement • Admins, courses leaders and module leaders can make track of student records and insert,update and alter student record if necessary. • Easy to create,upload necessary study materials for the students and student can view them easily.	• Moodle is not user-friendly and has basic UI interface • For managing Moodle technical knowledge is required. • Efficiency is lacking in Moodle.

• Moodle has separate notification panel	

2.1.3.1.2. Blackboard

Blackboard is web-based Course Management System. The main purpose of the blackboard course management system is to change the traditional way of learning and teaching. Likewise, other system Blackboard has also both positive and negative accepts which is discuss below.

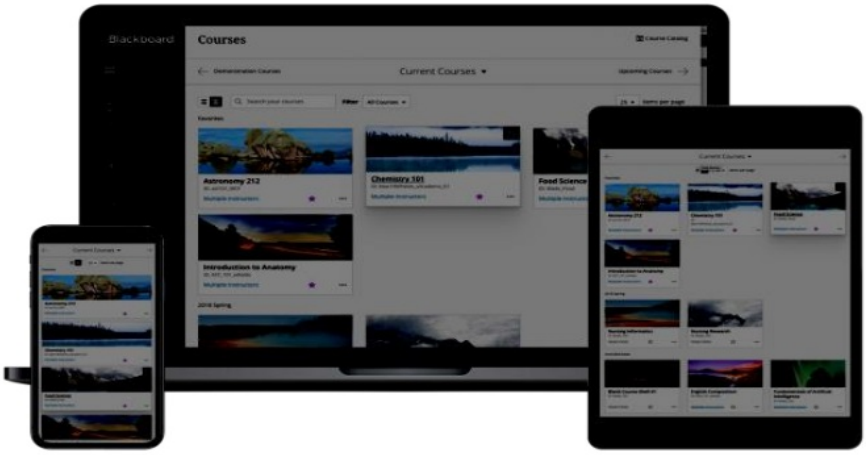


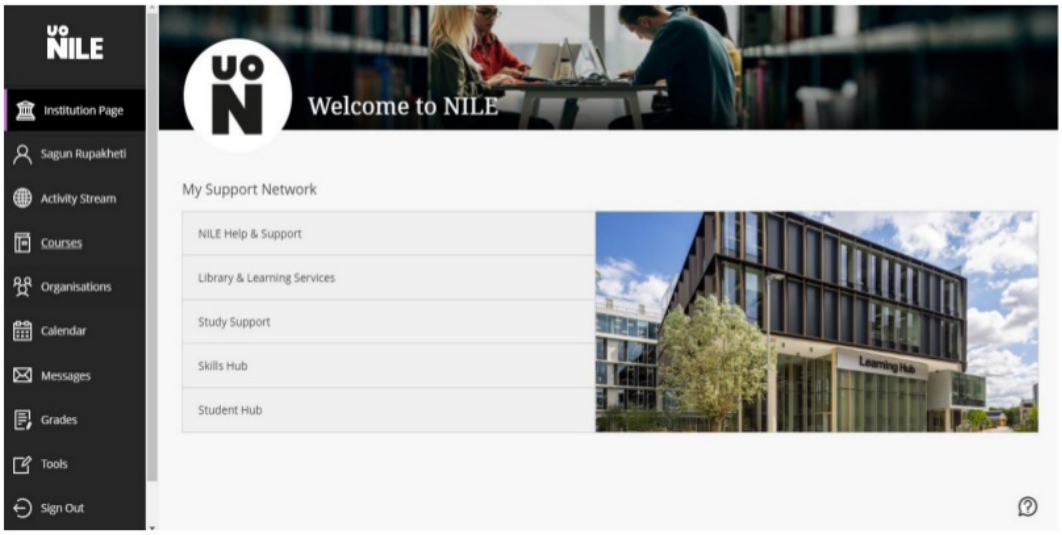
Figure 1. 2 Blackboard website

Positive aspects	Negative aspects
• Blackboard is user-friendly and has easy navigation system • For using Blackboard Technical training is not required • Low maintaining cost • Easy to create, update, alter and delete academic records such as course	• Blackboard required regular maintenance.

information, model activity and student inflammations	

2.1.3.1.3. NILE

NILE is the course management system of University of Northampton(UoN). It provides various functionalities that are very similar to the requirements of WUC.



Positive Aspects	Negative Aspects

• All the functionalities displayed in the navigation. • Neat and clean look. • Can edit own information • Can send messages to fellow students	• Too crowded navigation • Grades are not displayed

2.1.3.2 Development Relevant Legislation: -

2.1.3.2.1. Data Protection Act

The Data Protection Act (2018) is implemented by the UK. In this act it states that everyone or any organization that uses private data must follow strict rules called 'data protection principles. The user must be sure to use the data carefully, correctly, fairly, necessarily, accurately and securely.

The person whose data is being used must also be careful and must have stronger protection towards giving away their personal data. They must also be aware of the places, instances and ways their data is being collected and used.

If the data consists of private and more sensitive information such as information about race, ethnic background, political opinions, religious beliefs, trade union membership, genetics, biometrics, health and sex life of orientation then, the law must be strongly implemented.

2.1.3.2.2. Digital Economy Act

The Digital Economy Act (2017) is implemented by the UK. It states that the people and sectors are highly encouraged to be involved and digitalize their own sectors. The act states to empower consumers with quality product, provide better connectivity, build better infrastructures, enable better public services using digital technologies and provide citizens security from spam emails, scams, nudity and child pornography.

The digital economy is supposed to be clean and fruitful.

2.1.3.2.3. Computer Misuse Act

The Computer Misuse Act (1990) is implemented by the UK. With the easy access to the system through the internet there can be misuse of the system of for the unethical use of the application. The computer misuse act strictly prohibits to have unauthorized access to computer material with the intent to harm, harass, bully or rob. The internet must be used with all the ethics and precautions and there must be none relation to crimes like data stealing, online scam, robbery.

2.1.3.3 Any other relevant problem domain investigation data

There are many other that university was facing due to clerical based system. The other problem is discussed below

- Maintaining the student academic as well as personal records.
- Hard to track student admission record due to cercarial based system.
- Traditional way of teaching method which lead university to deliver quality of education.

- Tracking module activity is difficult for both module leader as well as student.
- Managing pat meeting for the student is difficult.

2.2. Requirements Specification

The application functional requirement and non-functional requirement is discussed in this section and system specification is made based on client requirement.

2.2.1. Problem Domain Description

The problem domain is an area of any problem that needs to be examined in order to solve the problem. Here, to solve the business limitations, the first step is to carry out the steps to determine all the problems. This section determines and explains all the problems and limitations in the existing clerical system of the college.

2.2.1.1. Existing Business Operation: -

Currently university is using traditional method for its day to day activity which is paper based. The current process of different day to day activity is discussed below.

2.2.1.1.1. UCAS and letter

A student applies to the college through UCAS. The UCAS sends a file with all the necessary information related to the student. The college now prepares a case paper based on information provided. If all the criteria is met, an unconditional letter is sent to the student. If the necessary criteria are not met, a conditional letter is sent to the student.

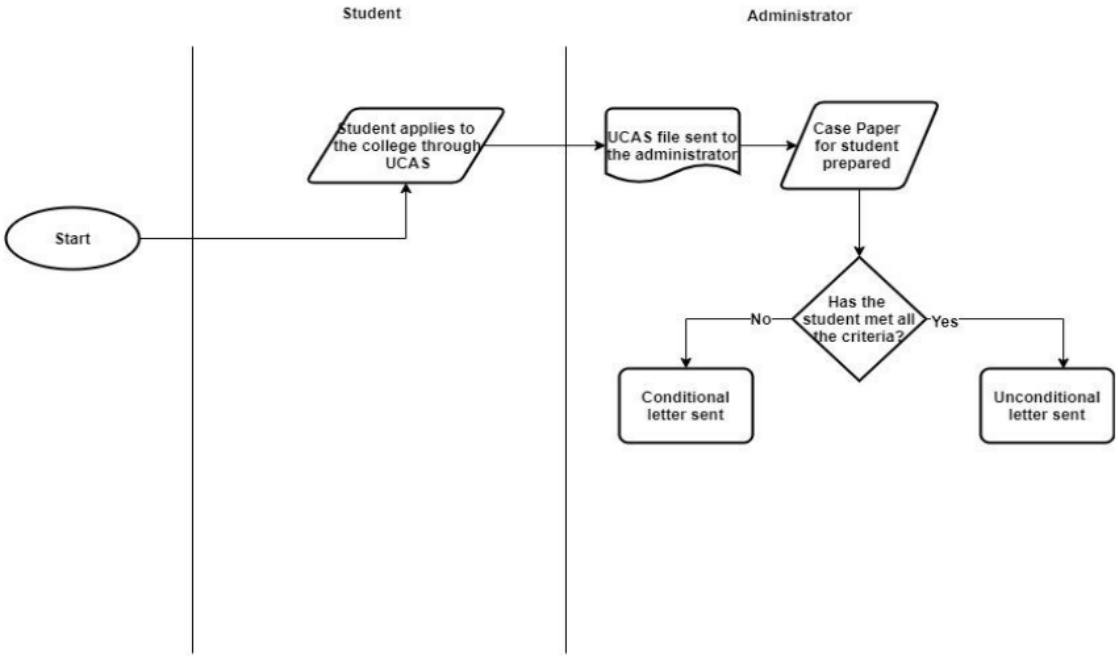


Figure 1. 3 Flowchart showcasing the ongoing process of admission through UCAS

2.2.1.1.2. Id generation and student folder

After the student has been classified as conditional or unconditional, an 8 digit identification code is generated. In a folder, the student's name and identification code is written and the case paper is added to the folder. The folder is stored in a place where all the case paper folders are stored.

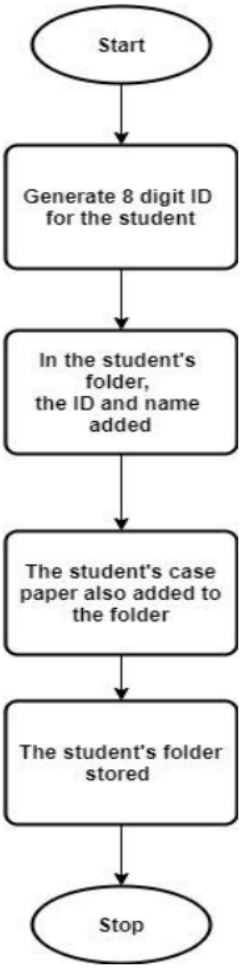


Figure 1.4 Flowchart showcasing the generation of Id and storage

2.2.1.1.3. Welcome Week

An email is sent to the student to attend the college for the welcome week. The student arrives at the welcome week. If the student has brought all the necessary documents, then the student is

admitted to the university and the provisional case paper is set to live. If not then the student is given a time span of 2 weeks to bring the documents. If after 2 weeks the student brings the documents then he/she is admitted else again a time period of another 2 weeks is given to the student. If the student fails to bring the documents even after the span of 4 weeks, the student is withdrawn and the status of the provisional case paper is set to dormant.

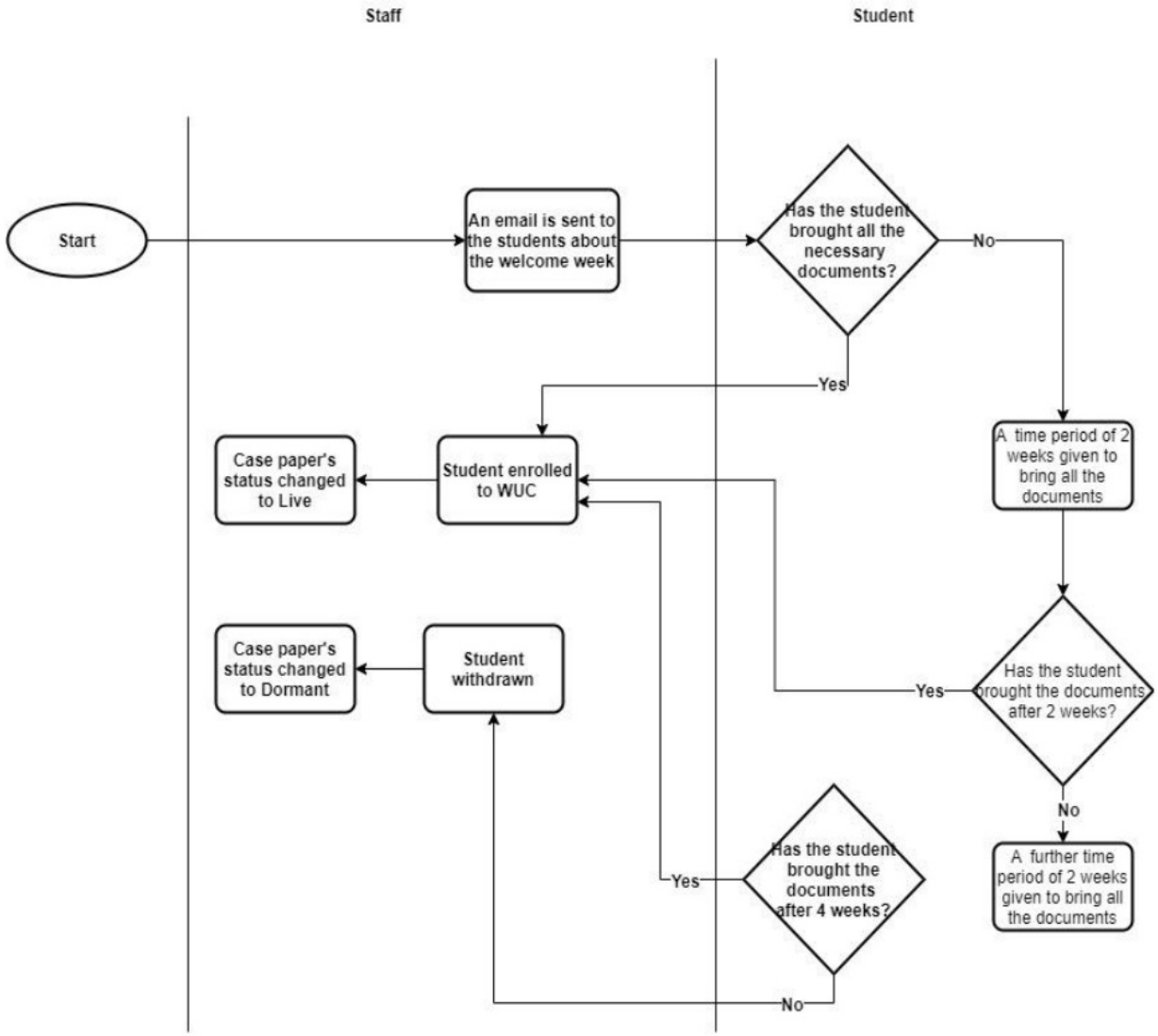


Figure 1. 5 Flowchart showcasing the welcome week and admission criteria

2.2.1.2. Summary of existing business limitations

The limitations give an overview of the sectors where the system needs the most help. All the processes suggest the following outcomes that need to be solved:

- The admission process is a lengthy and tedious process.
- Since the files are stored physically, they are vulnerable to being destroyed by fire or any other kind of damage.
- Since the announcements are physically put of somewhere in the notice boards, there is a high chance of missing out on the announcements.
- Assigning PAT to the students is a lengthy and tedious work and since there is no specific algorithm to assigning equal students to PATs , this might cause imbalance in the number of students a PAT has to deal with.
- There is no assurance of privacy of data.

2.2.2. Functional Requirements

This section covers the required functionalities of the proposed system. The functionalities proposed are supposed to solve all the existing problems in the college's current system. This portion of the report gives an overview about the access levels which are admin, module leader, students and their roles/ privileges in the system.

2.2.2.1. Required Features of the System

As all the current business limitations have been identified, the following features are considered appropriate solutions for the problems.

2.2.2.1.1. Database and backup

As the current system is a clerical based system, paper management has been very hectic and tedious. To overcome this problem the proposed system will completely store all the necessary information and data in the database which can be easily accessed at any given time. The problem of disposing all the dormant files after 10years of storage will be completely gone as the database will hold the dormant cases as long as the authorized people wish to hold onto the data in the system. This will also solve the issues of lack of storage spaces in the physical context.

2.2.2.1.2. Students records

Since the admission process of the existing system has a series of steps which turn out to be lengthy and inefficient, the system will control all the 3 stages/ status of the students based on their relation with the college/ course. The 3 particular statuses which will be dealt are provisional, live and dormant. Every student either currently enrolled or passed out or the ones that only applied to the university will have a particular status and their

data will be handled accordingly. Only the live students will have access to the all the features provided by the system.

2.2.2.1.3. Staff records

The system will hold and manage all the data related to the staffs that have ever been associated to the university. The concerned authorities will be able to add, edit or delete the information related to the staffs and the staffs will hold certain functionalities according to their designation. For example, an admin will be able to add, edit or delete other admin's information while a module leader will be able to add, edit or delete module's content as well as will be able to manage the visibilities of the content.

2.2.2.1.4. Course Records

- The administrators can add/edit/delete a course.
- A student must be enrolled in a certain course.
- Modules, levels and terms should be assigned to a course.

2.2.2.1.5. Module Management

- Administrators can add/edit/delete the modules.
- Students and module leaders can be assigned to all the modules.

2.2.2.1.6. Assignment Management

- The module leaders will be able add/edit/delete an assignment content.
- Module leaders can add the submission date and submission link.
- The students will be able to view and submit an assignment.

2.2.2.1.7. Attendance Management

- The module leaders can add the attendance information of any student.
- For a low attendance, the students are sent a warning message/email.
- The live student's information will also include the attendance percentage.

2.2.2.1.8. Personal Tutor Management

- A personal tutor is assigned to every student.
- The PAT and the students will be able to communicate in the system.

2.2.2.1.9. Grades Management

- The module leaders will be able to grade the assignments of the students.

- The grades will either be provisional or final.
- The module leaders can add feedbacks/ comments along with their grades.
- The students will be able to view their grades of each of their module.

2.2.2.1.10. Module Content Management

- The module leaders can add/edit/delete the module contents of their particular modules.
- Students can view/ download the files uploaded for the modules they are enrolled in.

2.2.2.1.11. Announcement Management

- The announcements can be added by the administrators
- The announcements for modules can be added by module leaders.
- Module announcements can be viewed only by the module leader and the students.

2.2.2.2. Access Rights/ Level of Access

This portion gives an overview about all the privileges given to different user groups. Some functionalities are common to the user groups while some vary. The administrators, module leaders and students all have different roles and are given different functionalities according to the necessity.

2.2.2.2.1. Administrator, Staff and Student Management

1. Administrator records management

Operation	Administrator	Module Leader	Student
Add Administrator	Add a new administrator	No access	No access
Edit Administrator	Amend the administrator's information	No access	No access
Delete Administrator	Delete the administrator completely	No access	No access
View Administrator	View an administrator	No access	No access
Archive Administrator	Archive administrator	No access	No access

Name of entity	Type/Format	Notes
User id	Integer(11 digits, no decimal place) Format- auto increment	Primary key for users

User first name	Integer(11 digits, no decimal place)	First name of the user
User middle name	String (up to 255 characters)	Middle name of the user
User surname	String (up to 255 characters)	Last name of the user
Password	Encrypted String (up to 255 characters)	Password to login
Birthdate	DATE	Date of birth
Contact	String (up to 255 characters)	Contact number
Address	String (up to 255 characters)	Current address
Gender	Enum(Male, Female)	Gender of the user
Email	String (up to 255 characters)	Email address of the user
Role	Enum(admin, student, module leader)	User's role

2. Module Leader records management

Operation	**Administrator**	**Module Leader**	**Student**
Add Module Leader	Add a new module leader	No access	No access

Edit Module Leader	Amend the module leader's information	No access	No access
Delete Module Leader	Delete the module leader completely	No access	No access
View Module Leader	View a module leader	View own profile	No access
Archive Module Leader	Archive module leader	No access	No access

Name of entity	**Type/Format**	**Notes**
Module leader id	Integer(11 digits, no decimal place) Format- auto increment	Primary key for module leaders (Composite key as also a foreign key from users)
Module id	Integer(11 digits, no decimal place)	Foreign key from modules
Experience	String(up to 255 characters)	The experience of the module leaders till date

2. **Student records management**

Operation	Administrator	Module Leader	Student
Add Module Leader	Add a new module leader	No access	No access
Edit Module Leader	Amend the module leader's information	No access	No access
Delete Module Leader	Delete the module leader completely	No access	No access
View Module Leader	View a module leader	View students profile	View own profile
Archive Module Leader	Archive module leader	No access	No access

Name of entity	Type/Format	Notes
Student id	Integer(11 digits, no decimal place) Format- auto increment	Primary key for students (Composite key as also a foreign key from users)
Course id	Integer(11 digits, no decimal place)	Foreign key from courses
GPA	Integer(11 digits)	The GPA of students
Previous college	String(up to 255 characters)	Previous college/school of students
Personal tutor id	Integer(11 digits)	Foreign key from users

Status	Enum(Provisional, Live, Dormant)	Status of students (provisional/live/dormant)
Level id	Integer(11 digits)	Foreign key from levels

2.2.2.2.2. Course Management

Operation	Administrator	Module Leader	Student
Add Course	Add a new course	No access	No access
Edit Course	Amend the courses' information	No access	No access
	Delete the course completely	No access	No access
View Course	View a course	No access	Can view the course in which the student is enrolled
Archive Course	Archive course	No access	No access

Name of entity	Type/Format	Notes
Course id	Integer(11 digits, no decimal place) Format- auto increment	Primary key for courses
Course name	String(up to 255 characters)	Name of the course

2.2.2.2.3. Module and Content Management

1. Module Management

Operation	Administrator	Module Leader	Student
Add Module	Add a new module in a course	No access	No access

Edit Module	Amend the module ‘s information	No access	No access
Assign Module	Assign a module to a module leader	No access	No access
Delete Module	Delete the module completely	No access	No access
View Module	View a module	No access	Can view the module in which the student is enrolled
Archive Module	Archive module	No access	No access

Name of entity	**Type/Format**	**Notes**
Module id	Integer(11 digits, no decimal place) Format- auto increment	Primary key for modules
Module code	String(up to 255 characters)	Code of the module
Module name	String(up to 255 characters)	Name of the module
Course id	Integer(11 digits, no decimal place)	Foreign key from courses

1. **Content Management**

Operation	**Administrator**	**Module Leader**	**Student**
Add Content	No access	Add a new content to a module	No access
Edit Content	No access	Amend the content	No access
Delete Content	No access	Delete the content completely	No access
View Content	No access	View the content	Can view the content of the module
Archive Content	No access	Archive content	No access

Name of entity	**Type/Format**	**Notes**

Content id	Integer(11 digits, no decimal place) Format- auto increment	Primary key for Contents
Term id	Integer(11 digits, no decimal place)	Foreign key from terms
Title	String(up to 255 characters)	Title of the content
Description	String(up to 255 characters)	Description of the content
Files	String(up to 255 characters)	File paths of the content
Visibility	Enum(Y, N)	Either to show or not show to the students

2.2.2.2.4. Assignment Management

Operation	Administrator	Module Leader	Student

Add Assignment	No access	Add a new assignment to a module for a term	No access
Edit Assignment	No access	Amend the assignment	No access
Delete Assignment	No access	Delete the assignment completely	No access
View Assignment	No access	View the assignment	Can view the assignment of the module for the term
Submit assignment	No access	No access	Can submit an assignment
Archive Assignment	No access	Archive assignment	No access

Name of entity	**Type/Format**	**Notes**
Assignment id	Integer(11 digits, no decimal place) Format- auto increment	Primary key for Assignments
Term id	Integer(11 digits, no decimal place)	Foreign key from Terms
Title	String(up to 255 characters)	Title of the Assignment
Description	String(up to 255 characters)	Description of the Assignment
Submission date	DATE	Date to submit the assignment

Files	String(up to 255 characters)	File paths of the Assignment
Visibility	Enum(Y, N)	Either to show or not show to the students

2.2.2.2.5. Grade Management

Operation	Administrator	Module Leader	Student

Add Grade	No access	Add a new grade to a module for a term for a student	No access
Edit Grade	No access	Amend the grade	No access
Delete Grade	No access	Delete the grade completely	No access
View Grade	No access	View the grade	Can view own grade of the module for the term
Archive Grade	No access	Archive grade	No access

Name of entity	**Type/Format**	**Notes**
Grade id	Integer(11 digits, no decimal place) Format- auto increment	Primary key for Grades
Student id	Integer(11 digits, no decimal place)	Foreign key from Students
Assignment id	Integer(11 digits, no decimal place)	Foreign key from Assignments

Published date	DATE	Date the grade had been published
Comments	String(up to 255 characters)	Any comments related to the grade
Visibility	Enum(Y, N)	Either to show or not show to the students

2.2.2.2.6. Attendance Management

Operation	Administrator	Module Leader	Student
Add Attendance	No access	Add attendance of a student	No access
Edit Attendance	No access	Amend the attendance	No access
Delete Attendance	No access	Delete the attendance completely	No access

Send warning	Send a warning if attendance is low	Send a warning if attendance is low	No access
View Attendance	View the attendance	View the attendance	Can view own attendance
View warning	View the warning	View the warning	View the warning
Archive Attendance	No access	Archive grade	No access

2.2.2.2.7. Personal Tutor Management

Operation	Administrator	Module Leader	Student
Assign Personal Tutor	Assign a personal tutor to a student	No access	No access

Edit Personal Tutor	Amend the personal tutor of a student	No access	No access
Delete Personal Tutor	Delete the personal tutor of a student	No access	No access
View Personal Tutor	View personal tutor	View themselves as a personal tutor	Can view own personal tutor
Archive Personal Tutor	Archive personal tutor	No access	No access

2.2.2.3. Output formats for onscreen displays

This portion deals with the onscreen displays which include all the emails sent which are triggered by some specific events. Various events lead to generation of certain format emails which are forwarded to the concerned user groups.

2.2.2.3.1. Enrolled to the system

Date: **Current Date**

Dear **Title First Name Last name,**

This is to bring you notice that you have successfully been enrolled to the Woodlands University College. Your user id is **ID** and password is and password is

- your date of birth in a six-digit date, month and year format
- the first two letters of your family name, with the first letter being a capital letter and the second letter being lower case
- the # sign

For example if your family name is Hobbs and your date of birth is 24 April 2000, your password would be: 240400Ho# .

You are suggested to reset the password as soon as you receive this email for security purpose. Please login through this portal **–LINK—**

We welcome you to our WUC whole heartedly. Thank you for being a part of our institution.

In case of any confusion or problem please contact **1234354654** or email at **sadsfs@woodlandsuni.ac.uk**

Note: Do not reply to this email but to the contacts given above.

2.2.2.3.2. Low Attendance

Date: **Current Date**

Dear **Title First Name Last name,**

This is to notify you that your attendance is **XX%** which is below the minimum attendance criteria in the **MODULE** as per informed by your module leader **MODULE LEADER**.

It is necessary that you meet me as soon as possible. We would like to know the cause of your failure to reach up to the minimum criteria and if possible come up with a solution to your problem. Your attendance plays a very important from your examination point of view as without the fulfilment of the criteria, you are not allowed to give the examination according to the rules of WUC which directly hampers your academic growth.

Please take this as an informal warning that your repetition of the module course is a possibility and meet me as soon as possible.

Best regards,

COURSE LEADER

Woodlands University College

2.2.2.3.3. Grades Published

Date: **Current Date**

Dear **Title First Name Last name,**

Your grades for the module MODULECODE have been published. Please go to the **LINK** to check your grades.

Check all your published grades at **LINK**

Best regards,

Woodlands University College

2.2.2.3.4. Assignment Published

Date: **Current Date**

Dear **Title First Name Last name,**

Your assignment content for the MODULE have been posted. Please go through the link [[LINK]] to go through the content.

Best Regards,

Woodlands University College

2.2.2.3.5. Assignment Submitted

Date: **Current Date**

Dear **Title First Name Last name,**

You have successfully submitted your **MODULE** assignment. Your submission link is **[[LINK]]** and your submission id is **123432**.

Best Regards,

Woodlands University College

2.2.2.3.6. PAT meeting schedule

Date: **Current Date**

Dear **Title First Name Last name,**

I have scheduled our weekly meeting for **DATE** at **TIME** in **VENUE**. Please make sure to be on time.

Best Regards,

PAT NAME

2.2.3. Performance Requirements

This portion of the report gives an idea about the pace and quality of the various functionalities of the system. It determines how quickly and how reliable the end-product system will be.

2.2.3.1. Speed

- **Throughput:** The throughput of the data will be greater and fetching the records will be a smooth process.

- **Response Time:** The target response time for the system will be around 4-6.5 seconds making it relatively a little higher than average response time. Nonetheless, around 10seconds of average speed would definitely be implemented.

2.2.3.2. Capacity

The capacity of the system is targeted initially to be around 2000 people. At least 2000 people will be able to access the system at the same time without having any lag in the response.

The amount of data and records to be stored initially is targeted to be around 60,000. However the capacity can always be expanded according to the need.

2.2.3.3. Reliability

The system will be available 24X7 a week i.e. for 168 hours excluding the time for maintenance and progress work for the system.

2.2.3.4. Usability

The system design (UI) will be of an easy and understandable level. Anyone can learn to use the system and finding functionalities will not be difficult as the navigation panel will give almost every sector for the user to get the necessary use out of the system.

2.2.4. Design Constraints

This portion explains the front end non-functional requirements that affect the system. It explains how the system is built not what the system does. The constraints have been imposed by the client himself and every constraint has been taken into consideration for building the system.

- **Target Operation System:**
  The system design will be compatible with every leading operating system in the market which are Windows, Linux, Android, etc.

- **Distributed or local architecture:**

  The architecture will be local to the university including extra storage for backups.

- **Required hardware requirements:**

The system will be able to work in any provided hardware either personal computers or mobile phones.

- **Front end graphic styles**

  The client has not given any specific color preference but, the system will follow the color combinations used in the logo of the university to maintain uniformity.

- **Programming languages**

  There are no specific programming language constraint by the client side, however the client has requested to use an object oriented approach (OOP) thus, the team has decided to use PHP as the base language for building the system. However, the code structure will strictly implement the MVC pattern for a systematic development.

- **Application Packages**

  For the storage of the data and records, MySQL will be used.

- **Development standards**

  The system will be built in the given time and given budget. The resources used for the system will be managed within the given budget.

- **Design approaches**

  The client has requested to use Object Oriented Design approach for the development of the system so OOP will be strictly implemented.

2.2.5. Commercial Constraints

This part of the report shows the breakdown of the given budget i.e. £20,000 used for various parts of the project development.

Basic calculations with estimated costs have been made keeping the project profit margin in consideration.

Project Information			
Total length of the project:	28 weeks		
Hourly Wage:	£18		
Number of Workers:	7		
Project Breakdown			
Sections	**Week**	**Hours per week per person**	**Total Wage**
Requirement specification	6	5	£1000.00
Design and analysis	7	5	£1000.00
Project build breakdown	12		
Console	25%	18	£1500.00
Website	30%	20	£2000.00

Mobile application	10%	20	£2000.00
Testing	30%	20	£2000.00
Evaluation	5%	20	£2000.00
Additional Cost			
Reason	**Cost**		
Software/ Hardware	£3000.00		
Office Costs()	£1500.00		
Total Build Cost			
Other expenses	£1000.00		
Profit Margin	£4500.00		
Program rights	Negotiable		
Total Project Cost	£17,000.00		

3. System Analysis & Design

The system analysis is considered as an important procedure for obtaining a better understanding about the system's required functionalities. As the brief demands to use an Object-Oriented approach for the development / completion of the system, the main areas to be covered for the correct analysis would be:

- Finding appropriate and necessary classes
- Proper implementation of the objects
- Interaction of the objects
- Operation of the objects
- The necessary internal procedures carried out by the objects

3.1. Preliminary Design Stages

To achieve the most efficient analysis of the scenario, a series of processes need to be carried out. There are many techniques as to which the potential classes can be identified within the system. The techniques below are considered ideal for the OOP approach as implementation of series of techniques ensure to give the optimum result for class identification.

3.1.1. Textual analysis

The textual analysis of the system is a method for candidate class identification. This method gives an idea that noun is equivalent to a candidate class and verbs are equivalent to the responsibilities possessed by the classes.

Through the principles of this method, a total of 3 classes could be identified which are admin, module leaders and students.

Candidate Classes	Responsibilities of the candidate classes
Admin	Add admin, edit admin, delete admin, add module leader, edit module leader, delete module leader, make student live, make student dormant, add announcement
Module Leaders	Add module content, edit module content, delete module content, add assignment content, edit assignment content, delete assignment content, add announcement, grade assignment
Students	view lecture, view assignment, view grades, view announcements, view attendance, upload assignment files

Courses	add course, edit course, delete course, view course
Modules	add module, edit module, delete module, view module
Assignments	add assignment, edit assignment, delete assignment, view assignment, submit assignment
Grades	add grade, edit grade, delete grade, view grade
Announcements	add announcement, edit announcement, delete announcement, view announcement
Module Contents	add content, edit content, delete content, view content, download content
Time-Table	add time-table, edit time-table, delete time-table, view time-table

3.1.2. Significant Event Analysis

The event analysis of the system includes different events of the system mapped along with the performers (classes) and candidate attributes of the performers. Basically, consideration of significant events help in the system design flow which leads to a proper and efficient mapping of the event flow of the software.

The events are said to be candidate class routines, performers include the candidate system classes and attributes contain candidate attributes for the candidate system classes.

Event	Performers	Candidate Attributes
Login	• Admin • Module Leader • Student	• User Id • Password • First name • Surname • Gender • Date of birth • Contact • Address • Email address • Course id • Module id • Level (student)
Logout	• Admin • Module Leader • Student	• User Id • Password • First name • Surname • Gender • Date of birth • Contact • Address • Email address • Course id • Module id

		• Level (student)
Add/edit/delete announcement	• Admin • Module Leader	• User Id • Password • First name • Surname • Gender • Date of birth • Contact • Address • Email address • Module id(module leader)
Add/edit/delete admin	• Admin	• Admin Id • Password • First name • Surname • Gender • Date of birth • Contact • Address • Email address • Course id
Add/edit/delete module leader	• Admin	• Admin Id • Password • First name • Surname

		• Gender • Date of birth • Contact • Address • Email address • Course id • Module Id
Add/edit/delete student	• Admin	• Admin Id • Password • First name • Surname • Gender • Date of birth • Contact • Address • Email address • Student Id • Course id
Add course	• Admin	• Admin Id • Password • First name • Surname • Gender • Date of birth • Contact

		• Address • Email address • Course id
Add modules	• Admin	• Admin Id • Password • First name • Surname • Gender • Date of birth • Contact • Address • Email address • Course id • Module id
Add/edit/delete module content	• Module Leader	• Module Leader Id • Password • First name • Surname • Gender • Date of birth • Contact • Address • Email address • Course id • Module id
Add/edit/delete assignment	• Module Leader	• Module Leader Id • Password • First name • Surname

		• Gender • Date of birth • Contact • Address • Email address • Course id • Module id • Assignment id
Grade assignment	• Module Leader	• Module Leader Id • Password • First name • Surname • Gender • Date of birth • Contact • Address • Email address • Course id • Module id • Assignment id • Student id
Add/edit/delete attendance	• Module Leader	• Module Leader Id • Password • First name • Surname • Gender • Date of birth • Contact

		• Address • Email address • Course id • Module id • Assignment id • Student id
View Announcement	• Admin • Module Leader • Student	• User Id • Password • First name • Surname • Gender • Date of birth • Contact • Address • Email address • Course id • Module id • Level (student)
View Grades	Student	• Student Id • Password • First name • Surname • Gender • Date of birth • Contact • Address • Email address • Course id

		• Module id • Level (student)
Request for information change	Student	• Student Id • Password • First name • Surname • Gender • Date of birth • Contact • Address • Email address • Course id • Module id • Level
View module content	Student	• Student Id • Password • First name • Surname • Gender • Date of birth • Contact • Address • Email address • Course id • Module id • Level
View assignments	Student	• Student Id • Password • First name

		• Surname • Gender • Date of birth • Contact • Address • Email address • Course id • Module id • Level • Assignment Id

3.1.3 Commands Queries and Constraints

The CQC technique is an efficient way of refining the classes produced by Significant Event Analysis or Textual Analysis. For each of the identified classes, a series of evaluation is supposed to be done under the specific headings Commands, Queries and Constraints. Commands specify the responsibilities of the class, Queries specify the attributes/functions of the class and Constraints specify the logical rules the class must align with.

CLASS	Module	Part: 1/1

TYPE OF OBJECT Add, edit delete and view module content		
Queries	Module Leader Id, Password , First name ,Surname ,Gender ,Date of birth ,Contact ,Address ,Email address ,Course id ,Module id ,Assignment id	
Commands	Add assignment, edit assignment, delete assignment	
Constraints	• Every assignment needs to have a brief. • Every assignment needs to have a submission date. • Every created assignment can be edited. • Every created assignment can be deleted.	

CLASS	Assignment	Part: 1/1

TYPE OF OBJECT Add, edit delete and view assignments		
Queries	Module Leader Id, Password , First name ,Surname ,Gender ,Date of birth ,Contact ,Address ,Email address ,Course id ,Module id ,Assignment id	
Commands	Add assignment, edit assignment, delete assignment	
Constraints	• Every assignment needs to have a brief. • Every assignment needs to have a submission date. • Every created assignment can be edited. • Every created assignment can be deleted.	

3.2 Detailed Static System Designs

At this stage, the system requirements is clear and entities for the system have been found out. By using the static system design approaches, the components for the system as well as designs are completed at the same time. The approaches used for static modelling are: BON diagrams, Use Case diagrams and Class diagrams. All approaches give a basic idea of the working of the system which makes it easier for the implementation process.

3.2.1. First Draft BON System Architecture Diagram

BON diagram (Business Object Notation diagram) is a method for analysis of any OO software. It is mainly used for the static modelling of the software.

Figure 1. 6 BON System Architecture Diagram

Figure 1. 7 BON System Architecture Diagram Flow

3.2.2. Use Case Diagrams

Use case diagrams are a form of static design representation. They showcase how a proposed system must look from the user's perspective. They capture the functional requirements of the system and must capture the user's goals not the systems' functions.

3.2.2.1. Administrator Use Case Diagram:

Below is the use case representation of the functionalities performed by the administrator in the system. The administrator has the highest level of access and moderation in the system.

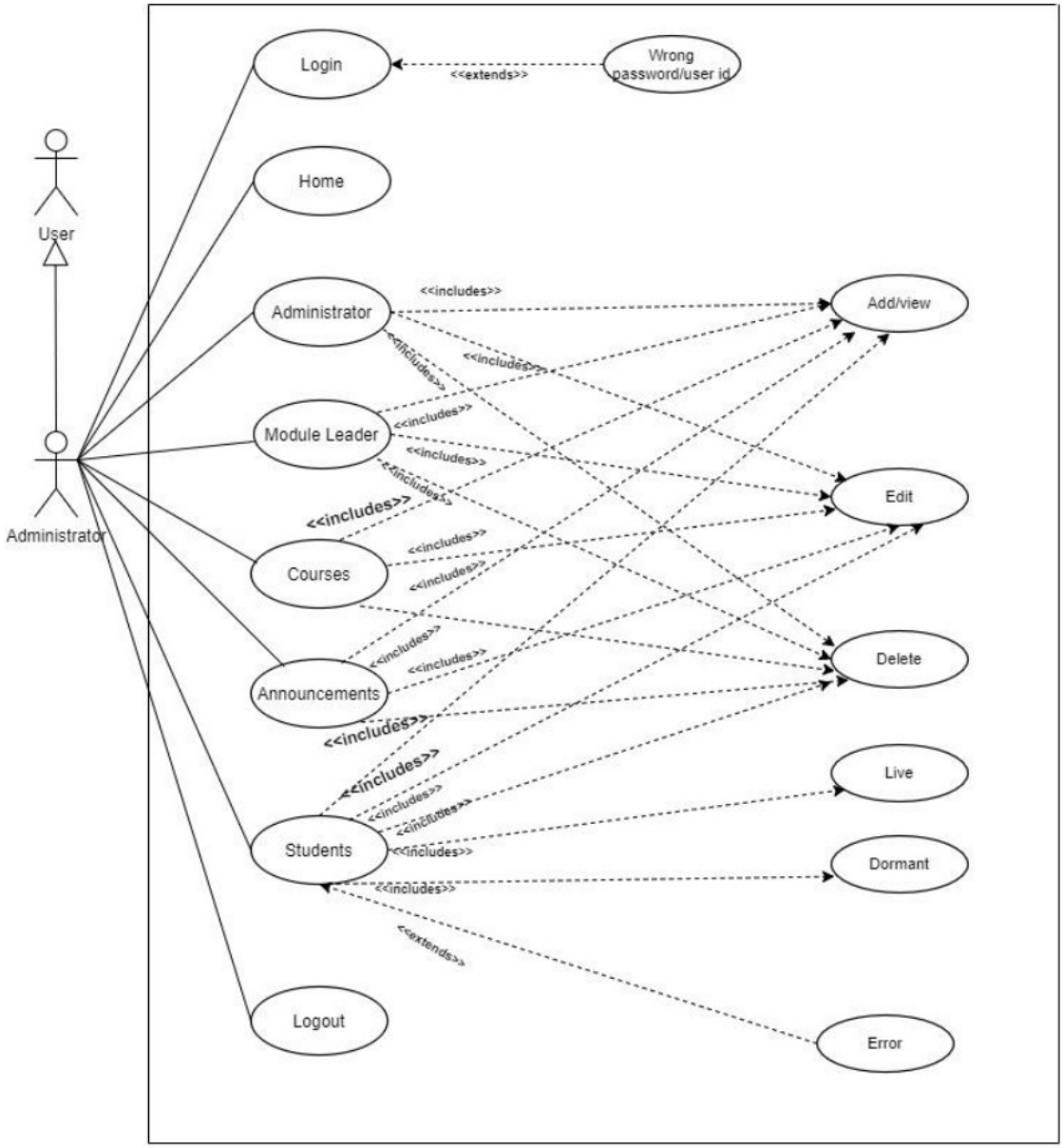


Figure 1. 8 Admin Use Case Diagram

3.2.2.2. Module Leader Use Case Diagram

Below is the use case representation of the functionalities performed by the module leader. The module leader has the maximum authority towards their own module which includes adding, editing and deleting module contents.

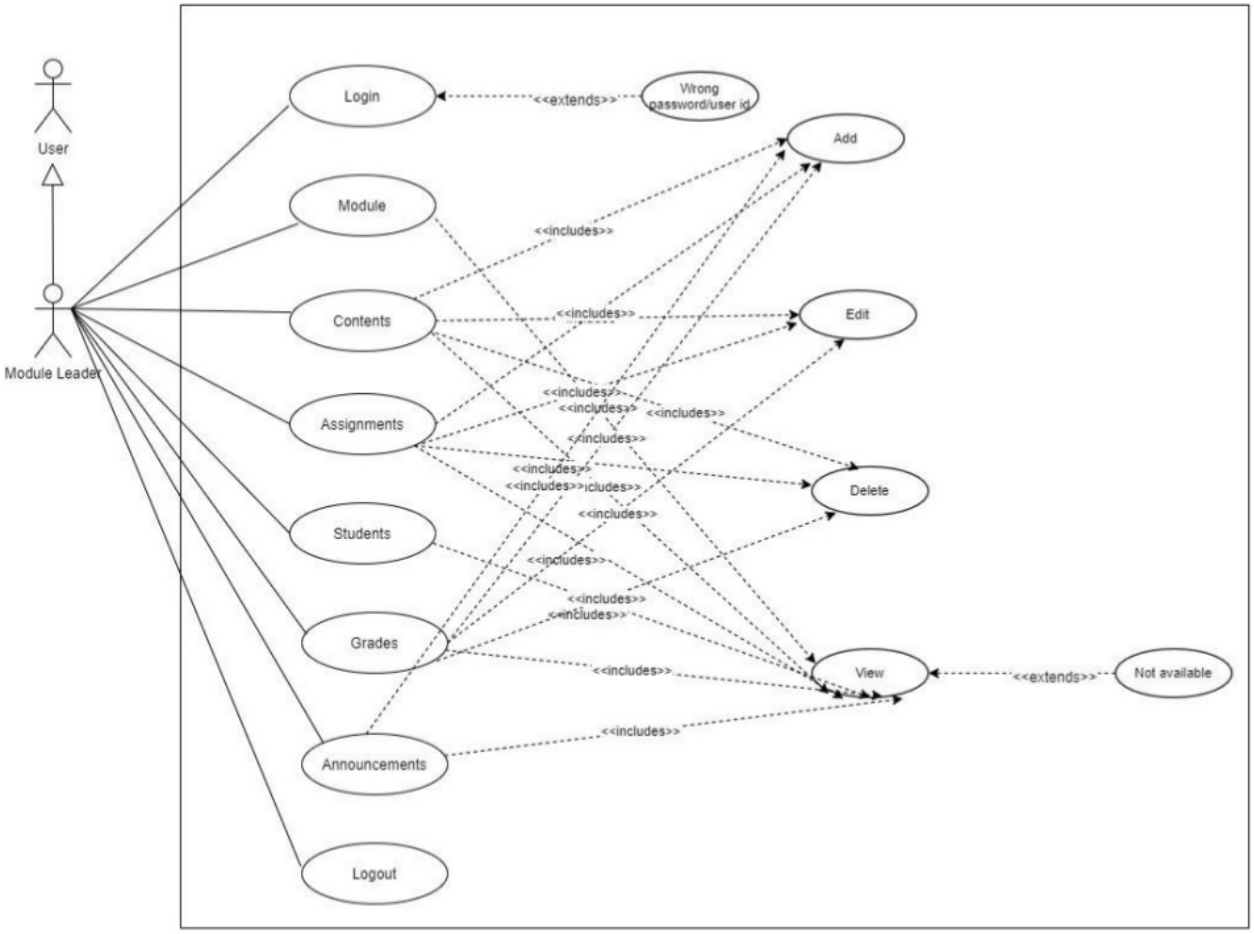


Figure 1. 9 Module Leader Use Case Diagram

3.2.2.3. Student Use Case Diagram

Below is the use case representation of the various functionalities of the student in the system. The functionalities include viewing their courses, modules, uploading assignments and viewing grades.

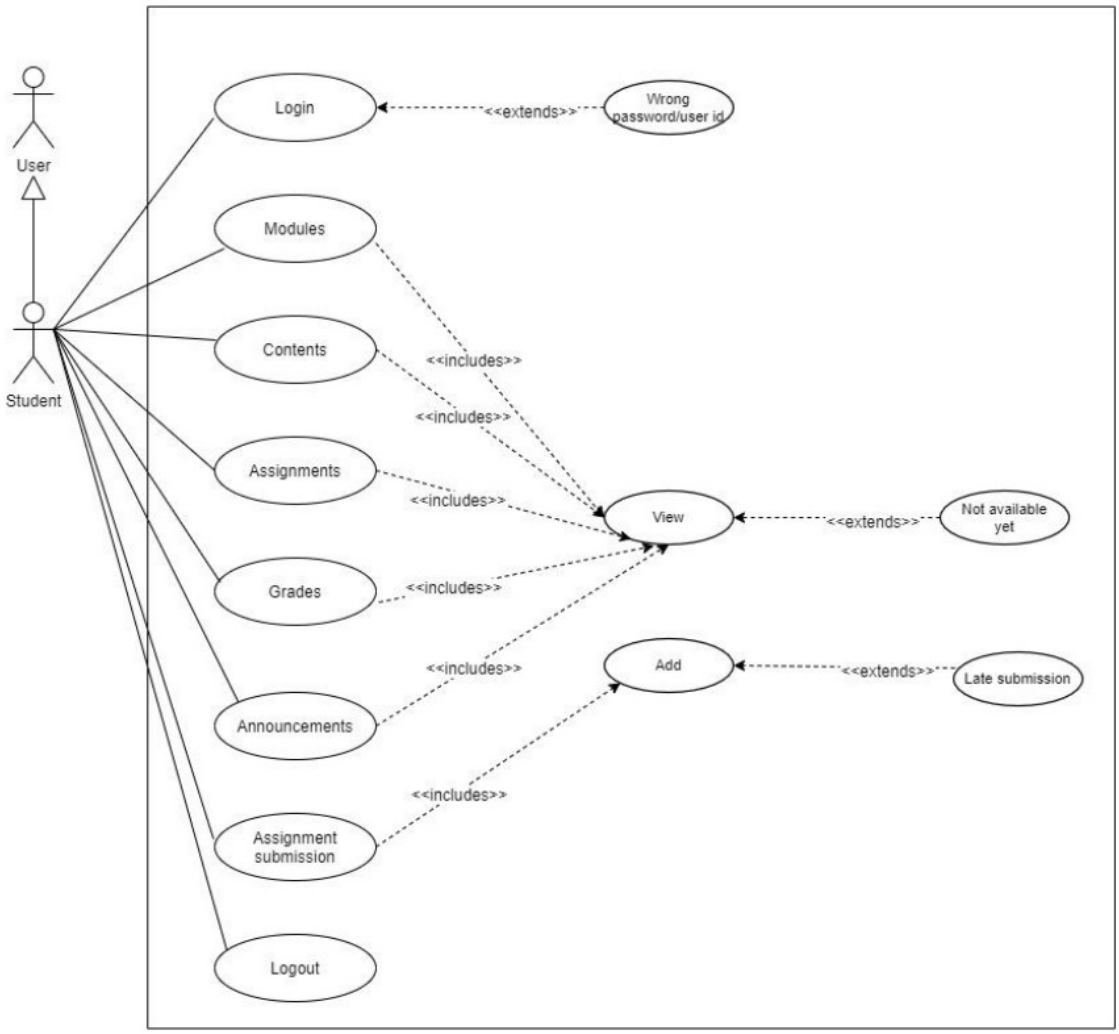


Figure 1. 10 Student Use Case Diagram

3.2.3. Use case documentations

This portion shows the documentation for the use case diagrams above. The documentation has been done only for the administrator.

3.2.3.1. UC1 Administrator tab

Idenfier and name: UC1 Administrator tab

Initiator: Administrator

Goal: To perform actions on the administrators

Pre-condition: No action has been taken

Post-condition: The administrator performs the required action on other adminstrators

Assumptions: The administrator is logged in

Main success scenario:

1. Administrator clicks on the Administrator tab.
2. The adminstrator chooses to add/ update / delete the admin's information.
3. If no errors then the required action is completed.
4. The system either adds or updates or deletes the admin's information based on the action.

3.2.3.2. UC2 Module Leader tab

Idenfier and name: UC2 Module Leader

Initiator: Administrator

Goal: To perform actions on the module leaders

Pre-condition: No action has been taken

Post-condition: The administrator performs the required action on module leaders

Assumptions: The administrator is logged in

Main success scenario:

1. Administrator clicks on the Module Leaders tab.
2. The adminstrator chooses to add/ update / delete the module leader's information.
3. If no errors then the required action is completed.
4. The system either adds or updates or deletes the module leader's information based on the action.

3.2.3.3. UC3 Announcements tab

Idenfier and name: UC4 Courses

Initiator: Administrator

Goal: To perform actions on the courses

Pre-condition: No action has been taken

Post-condition: The administrator performs the required action on the courses of the college

Assumptions: The administrator is logged in

Main success scenario:

1. Administrator clicks on the Courses tab.
2. The adminstrator chooses to add/ update / delete the courses/ courses' information.
3. If no errors then the required action is completed.
4. The system either adds or updates or deletes the course information based on the action.

3.2.3.4. UC4 Courses tab

Idenfier and name: UC4 Courses

Initiator: Administrator

Goal: To perform actions on the courses

Pre-condition: No action has been taken

Post-condition: The administrator performs the required action on the courses of the college

Assumptions: The administrator is logged in

Main success scenario:

1. Administrator clicks on the Courses tab.
2. The adminstrator chooses to add/ update / delete the courses/ courses' information.
3. If no errors then the required action is completed.
4. The system either adds or updates or deletes the course information based on the action.

3.2.3.5. UC5 Students tab

Idenfier and name: UC5 Students

Initiator: Administrator

Goal: To perform actions on the students

Pre-condition: No action has been taken

Post-condition: The administrator performs the required action on the students registered

Assumptions: The administrator is logged in

Main success scenario:

1. Administrator clicks on the Courses tab.
2. The adminstrator chooses to add/ update / delete the courses/ courses' information.
3. If no errors then the required action is completed.
4. The system either adds or updates or deletes the course information based on the action.

3.2.4. Class diagram

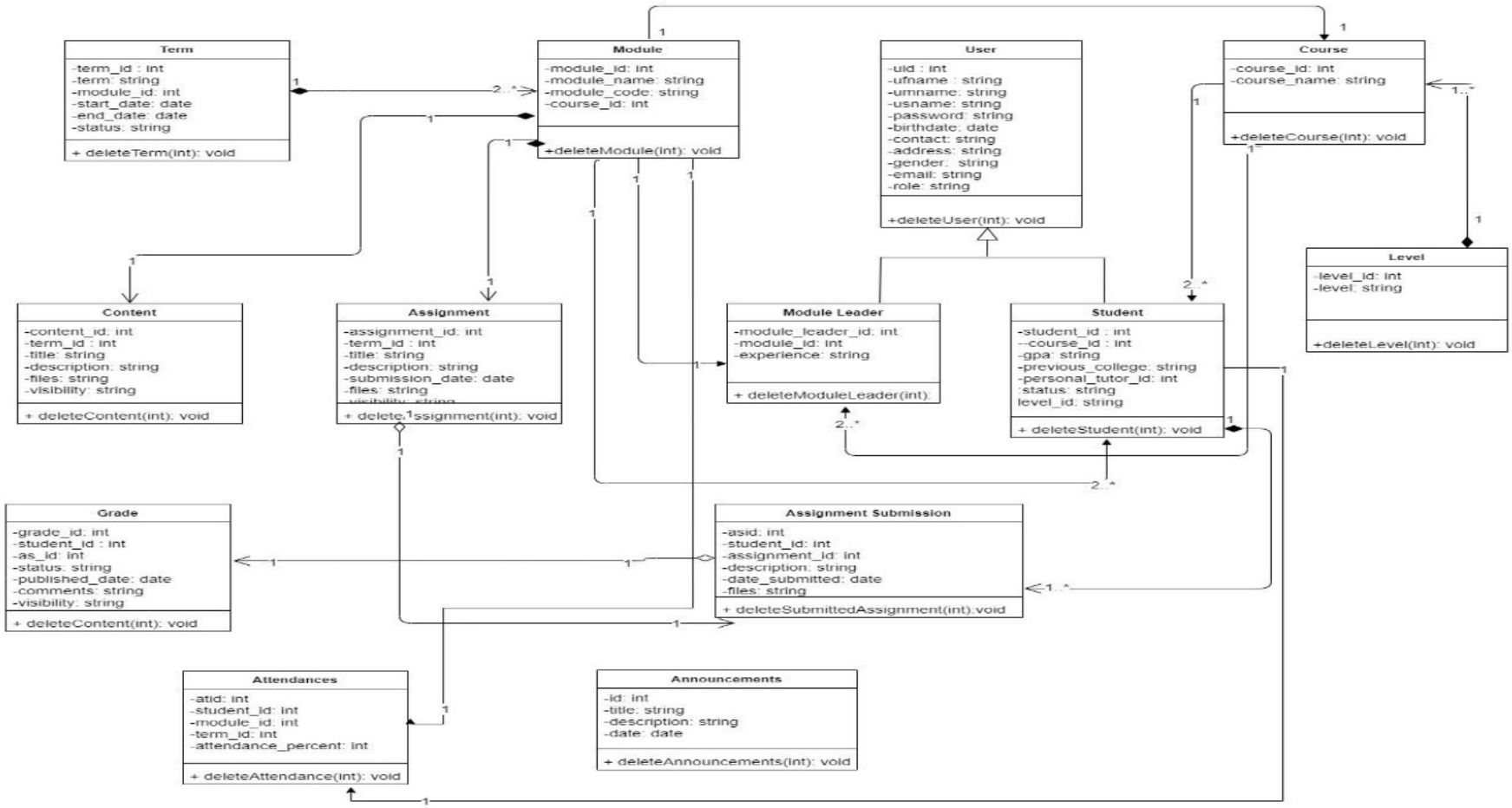


Figure 1. 11 System Class Diagram

Above is the class diagram for system. Class diagram is basically the representation of the system flow based on various classes used in the system. Classes are collection of objects with common structure and behavior. The relationships used in the diagram below include both composition and aggregation relationships.

3.3 Detailed Dynamic System Designs

This portion consists of the dynamic view of the system. Unlike the static design, the dynamic designs represent the system flow in the more proper manner showcasing the use of methods.

3.3.1. System Sequence Diagrams

Sequence diagrams are basically the graphical representation of use case diagrams. They show the interactions between the use case and the objects. The sequence diagram consist of a lifeline that shows the instance of an actor or a object. The activation box is attached to the lifeline that shows the period of time the object is performing certain task. The object/ actor is said to have live activation at this time. The arrow represents the message in the form of methods.

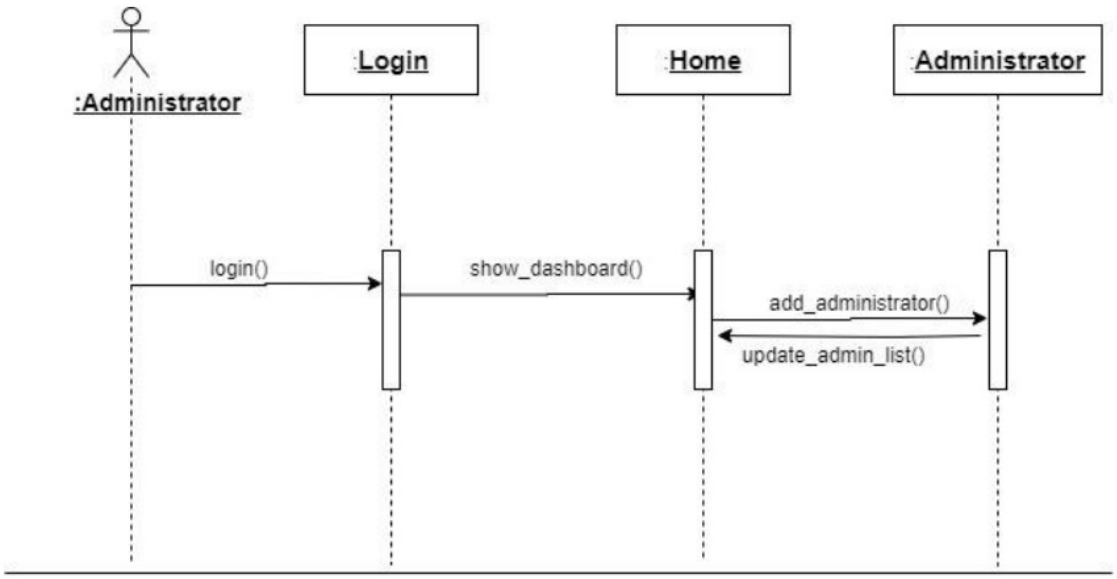


Figure 1. 12 Sequence diagram for adding an administrator

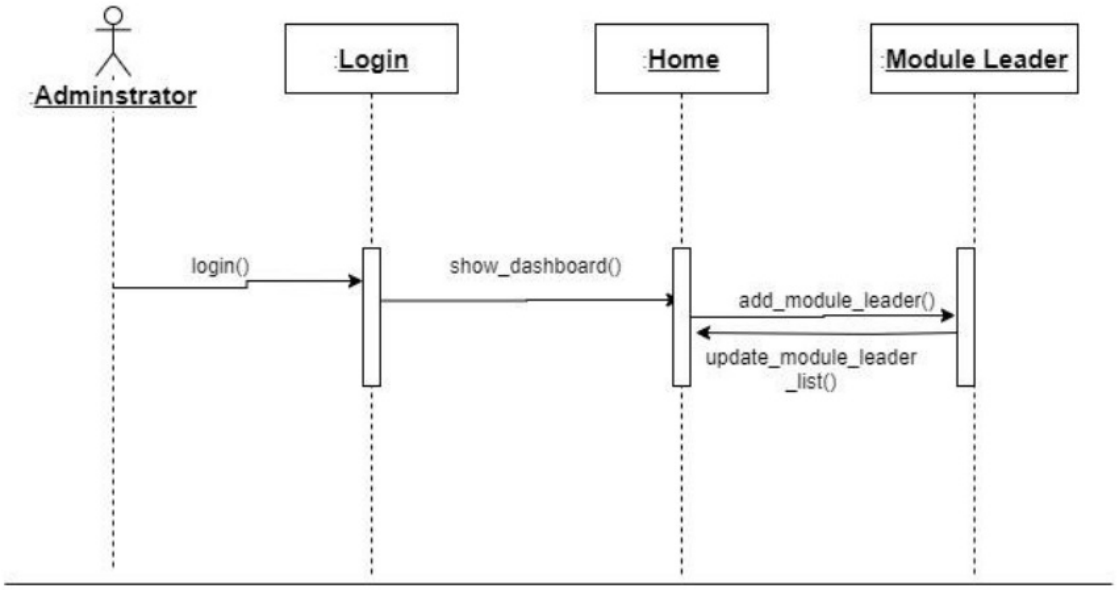


Figure 1. 13 Sequence diagram for adding a module leader

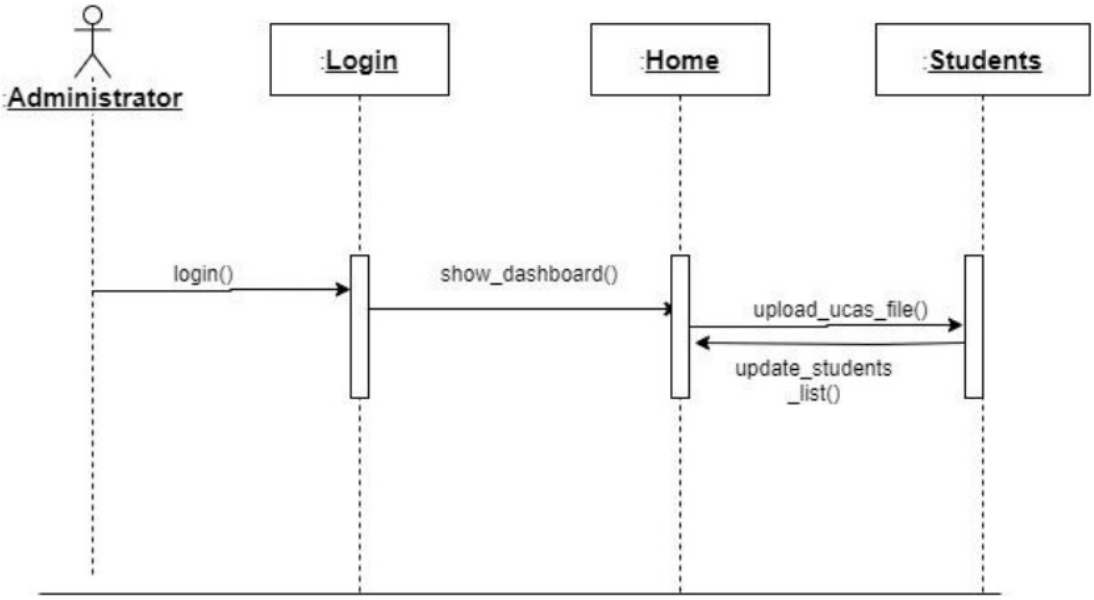


Figure 1. 14. Sequence diagram for adding a student to the system

3.3.2. System Collaboration/ Communication Diagrams

The system collaboration diagrams are similar to the sequence diagrams. The only difference between them is that communication diagrams show the link between the objects and actors without any activation boxes or lifelines.

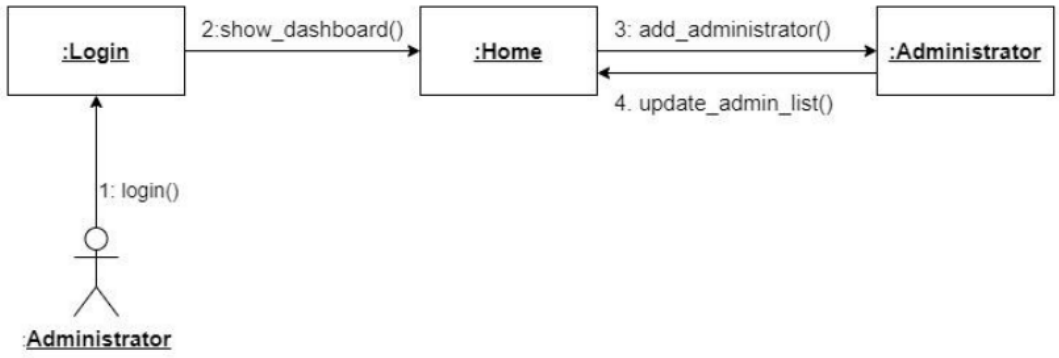


Figure 1. 15 Collaboration diagram for adding an administrator

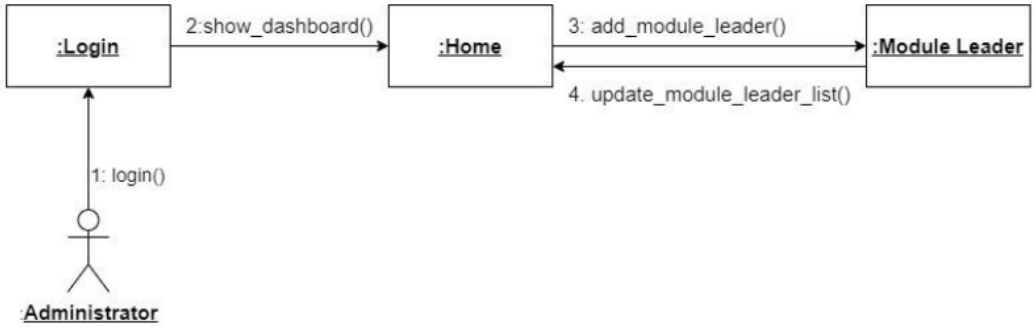


Figure 1. 16 Collaboration diagram for adding a module leader

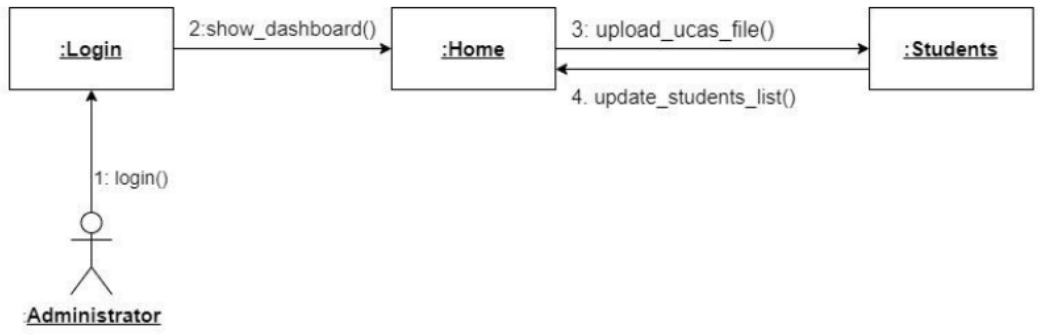


Figure 1. 17. Collaboration diagram for adding a student to the system

3.4. System Database Design

The database of the system has developed as the project development started. Necessary changes and amendments have been made throughout the project development process.

3.4.1 E-R Model

The figure below shows the tables and their relationships for the system.

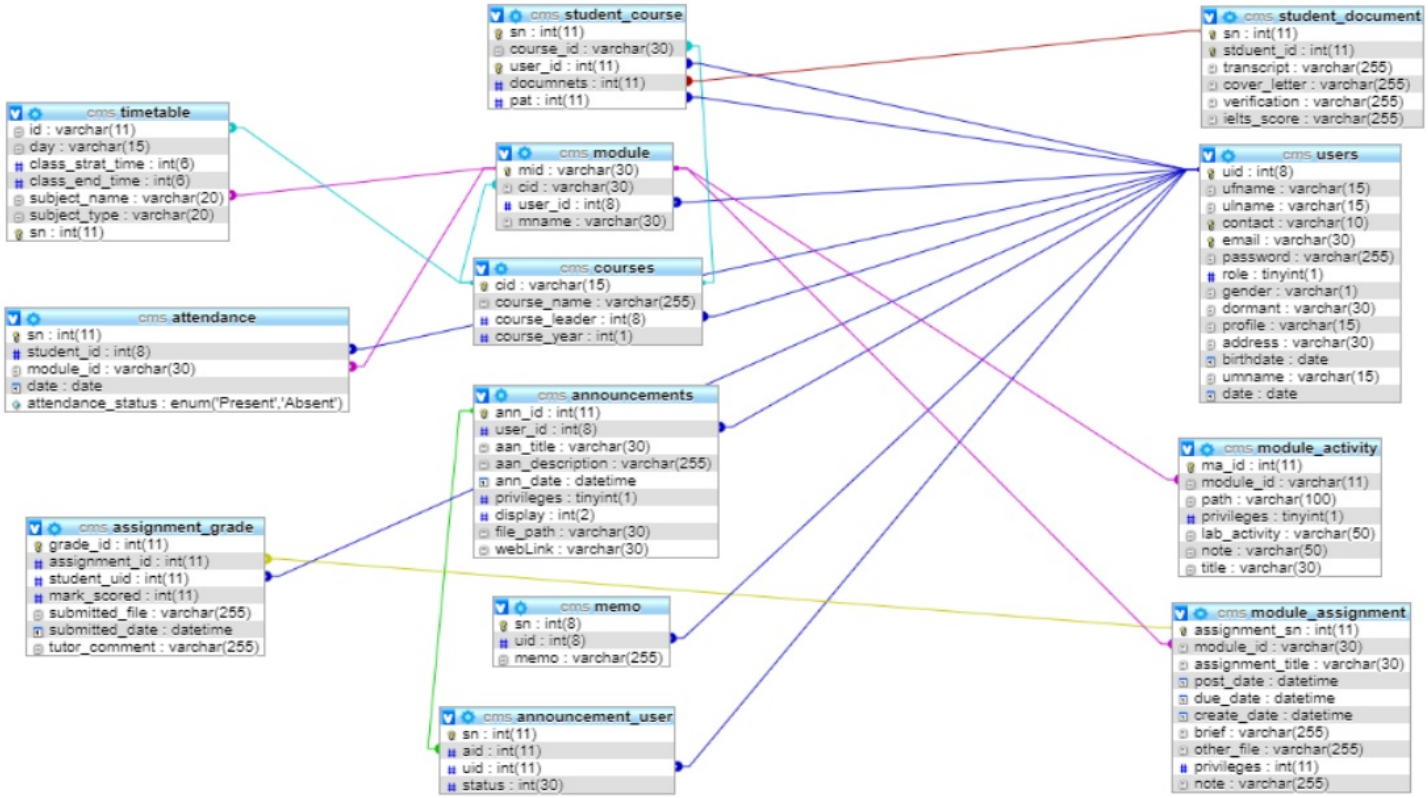


Figure: E-R view

3.4.2 Attribute Listings

The table below showcases all the attributes for all the tables in the database.

S.no.	Table Name	Attributes
1.	users	uid, Ufname, umname, ulname,contact, email, password, role, gender, dormant, profile, address, birthdate, date
2.	timetable	id,day, class_start_time, class_end_time, subject_name, subject_type, sn
3.	student_document	sn, student_id, transcript, cover_letter ,verification, ielts score
4.	student_course	sn, course_id, user_id, documents, pat
5.	module_assignment	assignment_sn, module_id, assignment_title, post_date, due_date, create_date, brief, other_file, privileges, note
6.	module_activity	ma_id, module_id, path, privileges, lab_activity, note, title
7.	module	mid, cid, user_id, mname
8.	memo	sn, uid, memo
9.	courses	cid, course_name, course_leader, course_year
10.	attendance	sn, student_id, module_id, date, attendance_status
11.	assignment_grade	grade_id, assignment_id, student_uid, mark_scored, submitted_file, tutor_comment

12.	announcement_user	sn, aid, uid, status
13.	announcements	ann_id, user_id, aan_title, aan_description,ann_date, privileges, display, file_path, webLink

4. System Interface Designs

This portion consists of visualization of the view of the system. Different methodologies are used in order to make the implementation process easier. The processes used are draft interface designs, design revisions and heuristic evaluation.

4.1. Draft Interface Designs

The draft interface designing is done by the processes wireframes, system mock-ups, activity event diagrams, navigation diagrams.

4.1.1. Wireframes

Wireframes are the most initial phase to building a website/ system. It is a simple sketch which provides a basic idea of how system's various pages look like and behave. The wireframes consist of specific details which are important to the client.

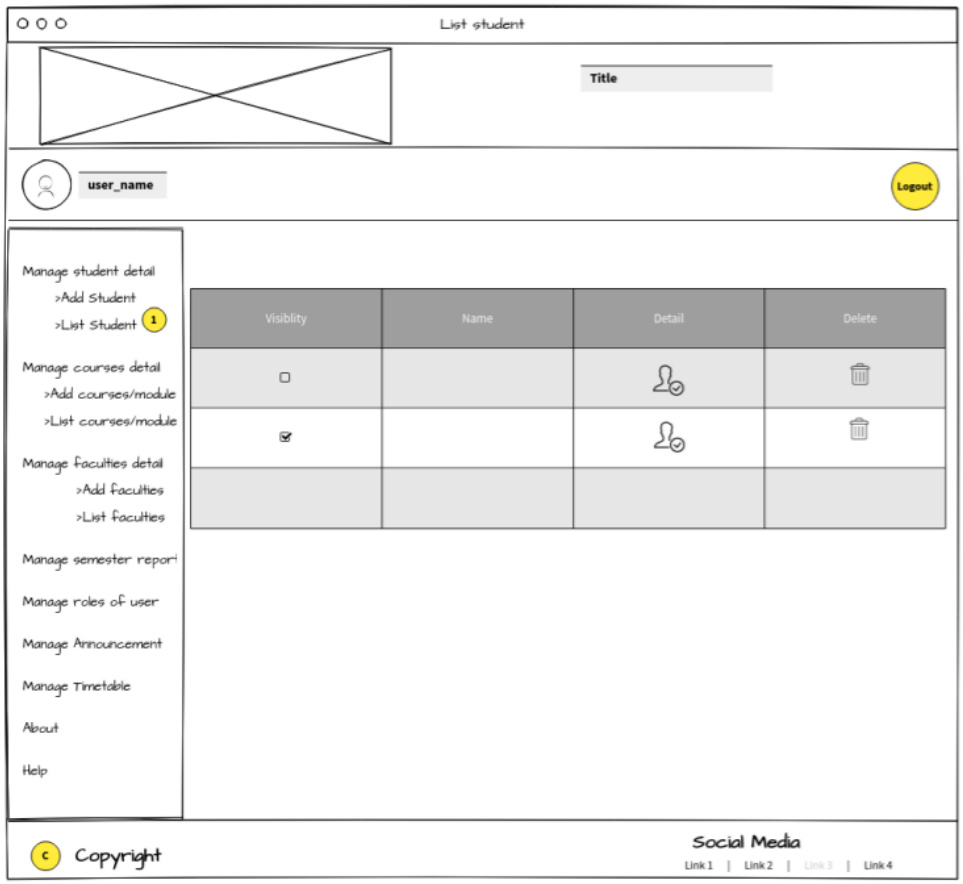


Figure 1. 18. Wireframe showing students list

Figure 1. 19 Wireframe showing student's details

4.1.2. System Navigation Diagram

The system navigation diagrams show the flow of the system. It basically portraits the various links that connect the system and make the working of the system functional. It aims to provide an overview of all system screens and show their pathways.

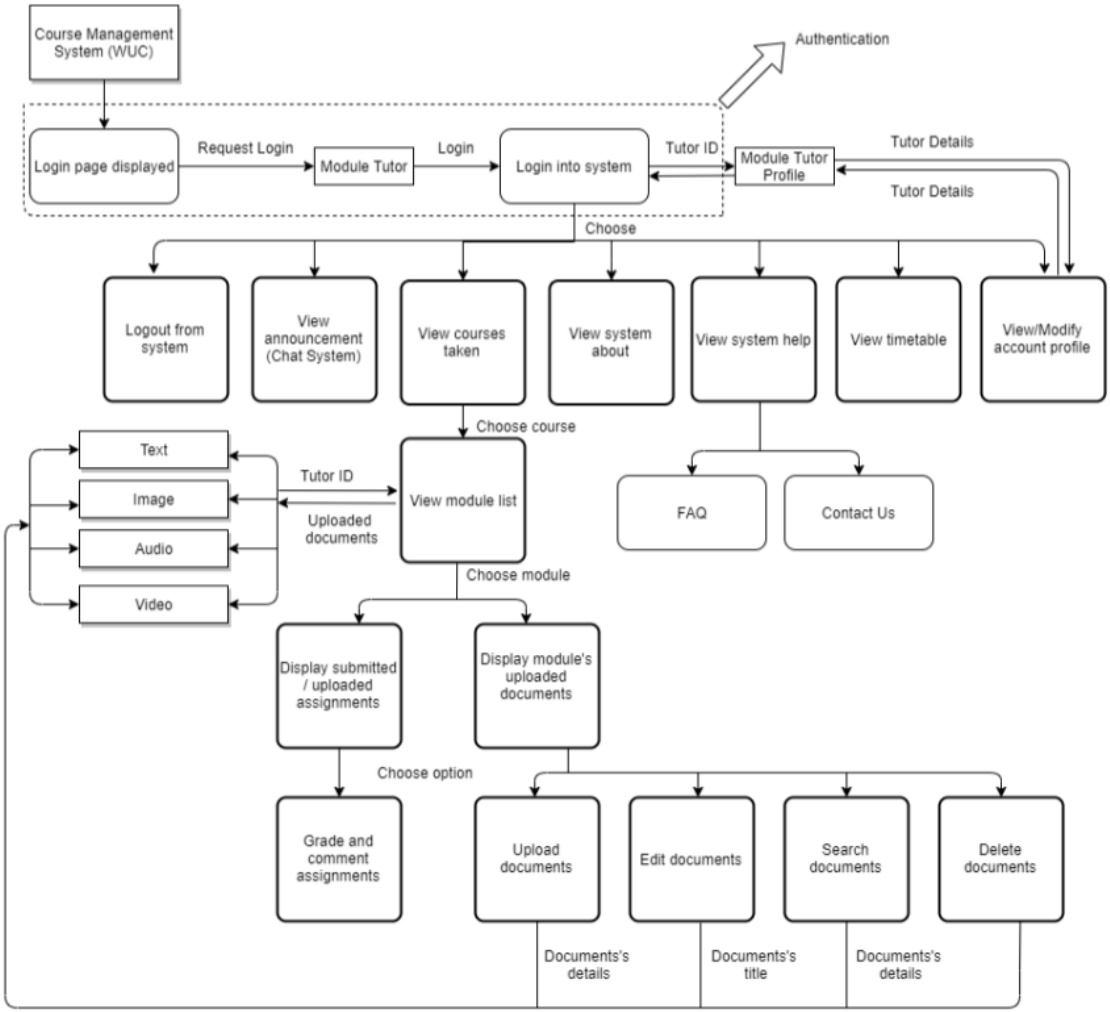


Figure 1. 20. Detailed navigation of module leader

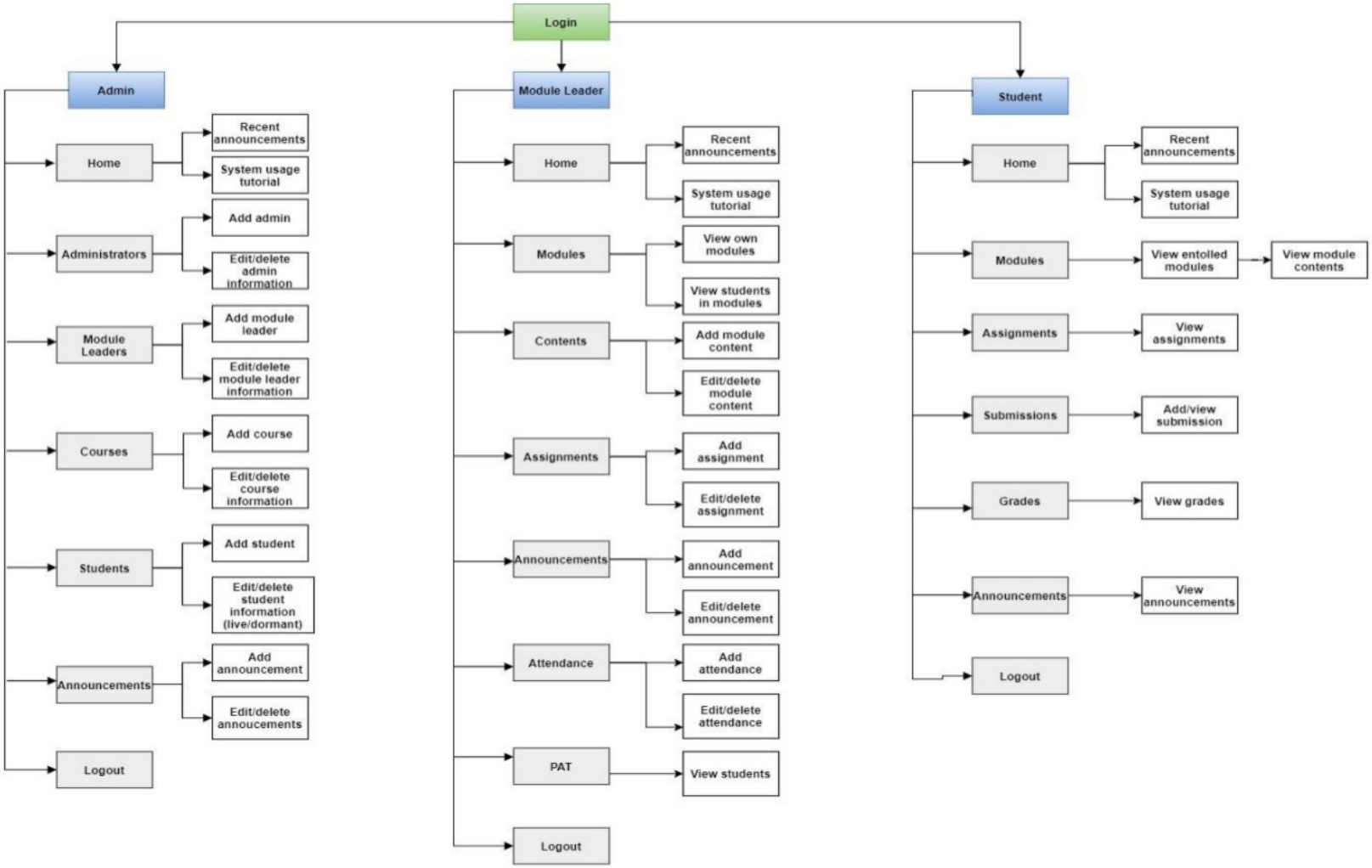


Figure 1. 21 Full system navigation diagram

4.1.3. System Screen mock-ups

System mockups are the most efficient in visualizing a system. It shows the actual representation of the system according to the finalized view. Even though mockups are visual representations, they only give the basic idea of the system and might be a lot different than the actual system.

Admin

Figure 1. 22. Admin Dashboard

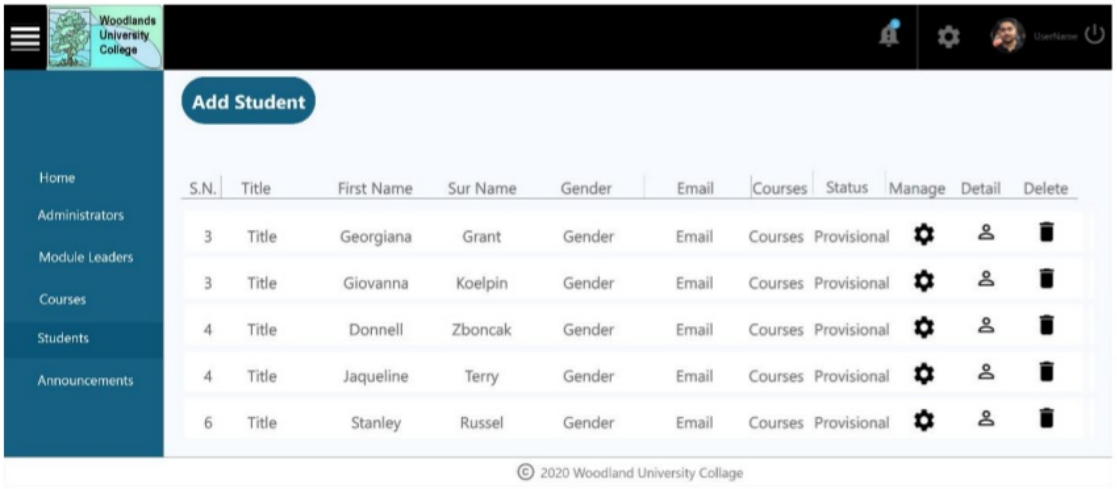


Figure 1. 23 The admin clicks on Students tab

4.1.4. System Activity Event Diagrams

The system activity diagrams shows the flow of the system in the form of flowcharts. The flowchart represents the activity for specific events.

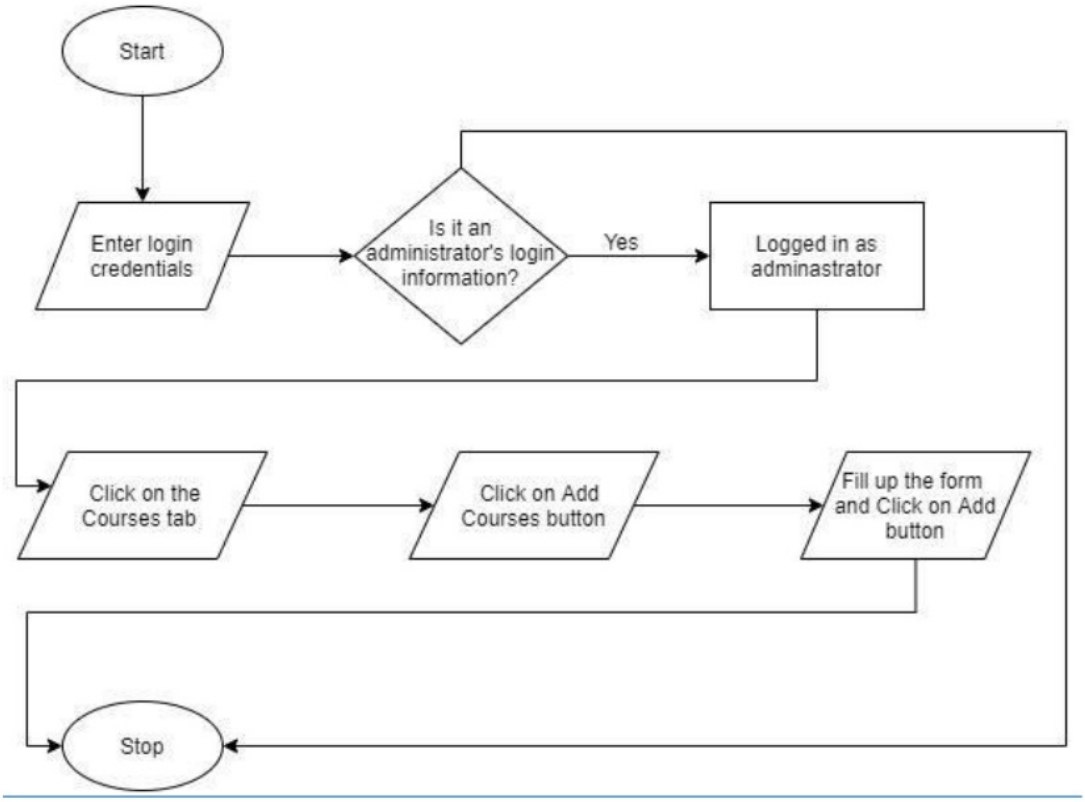


Figure 1. 24. Flowchart showing the process of adding the course

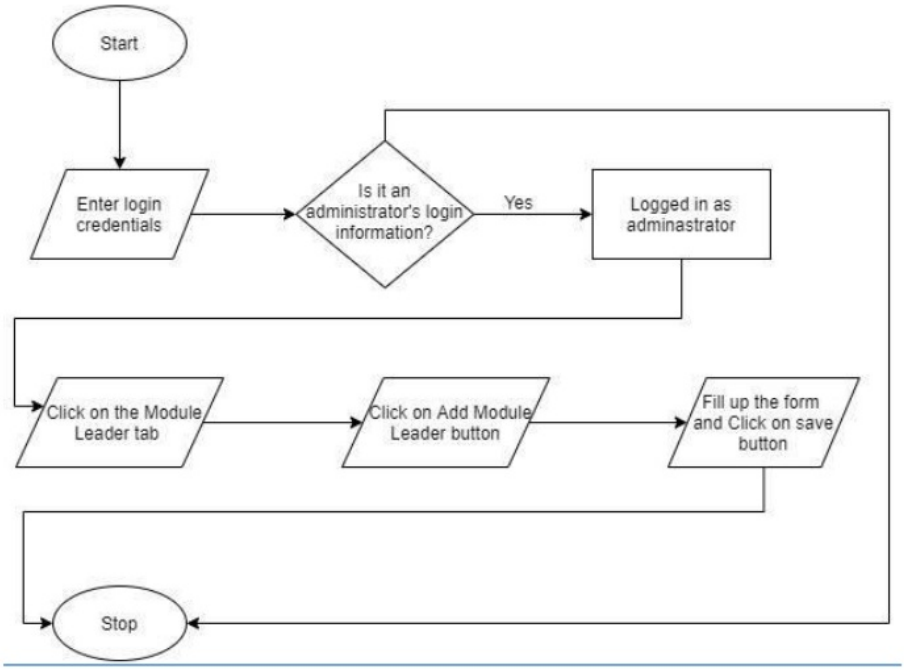


Figure 1. 25. Flowchart showing the process of adding a module leader

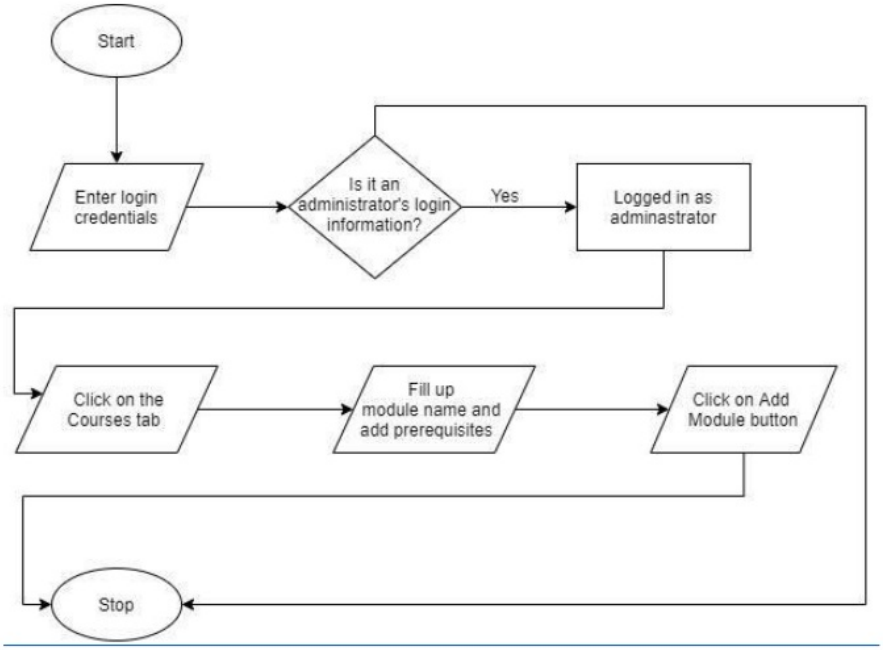


Figure 1. 26 Flowchart showing the process of adding a module

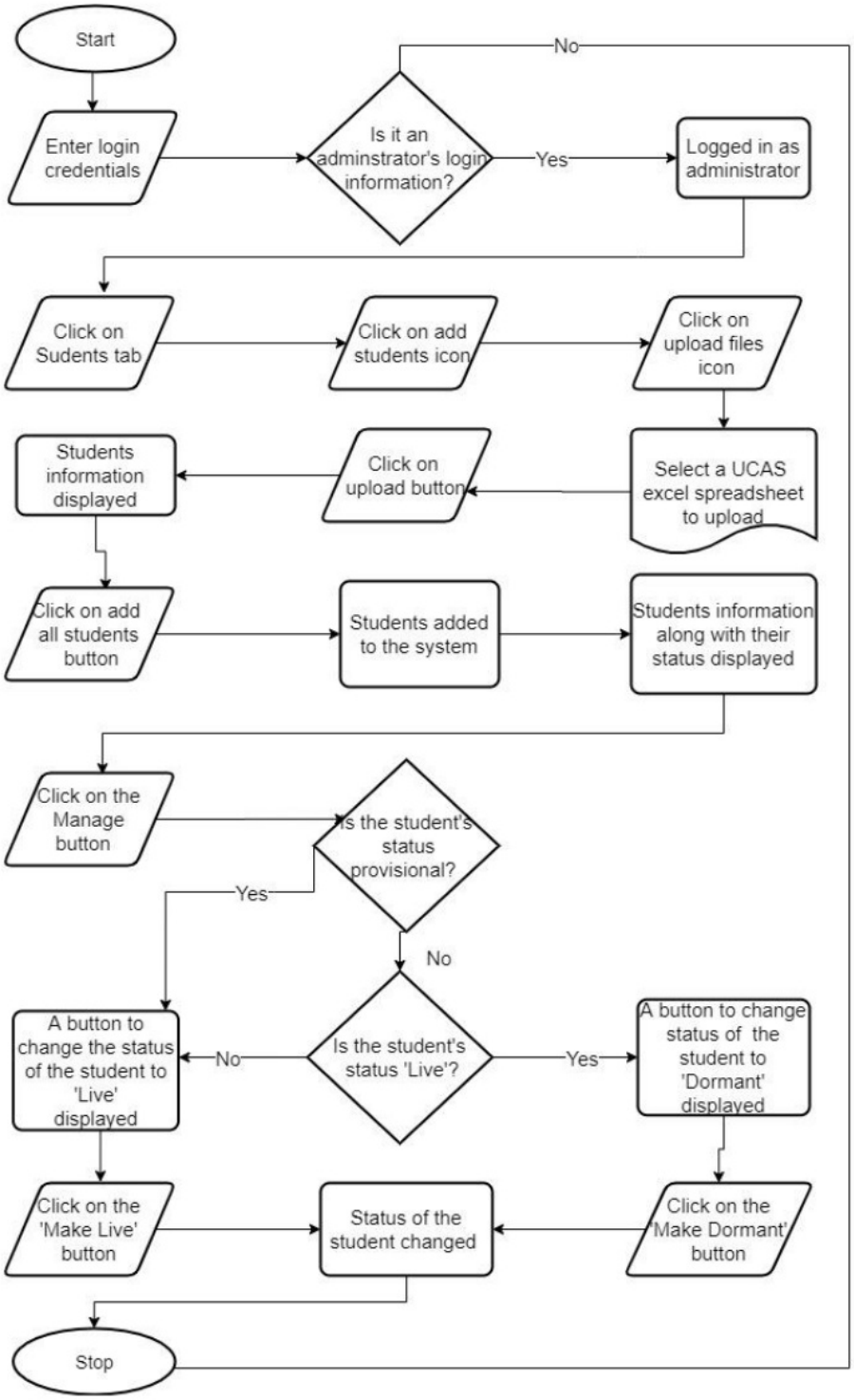


Figure 1. 27. Flowchart showing the admission process

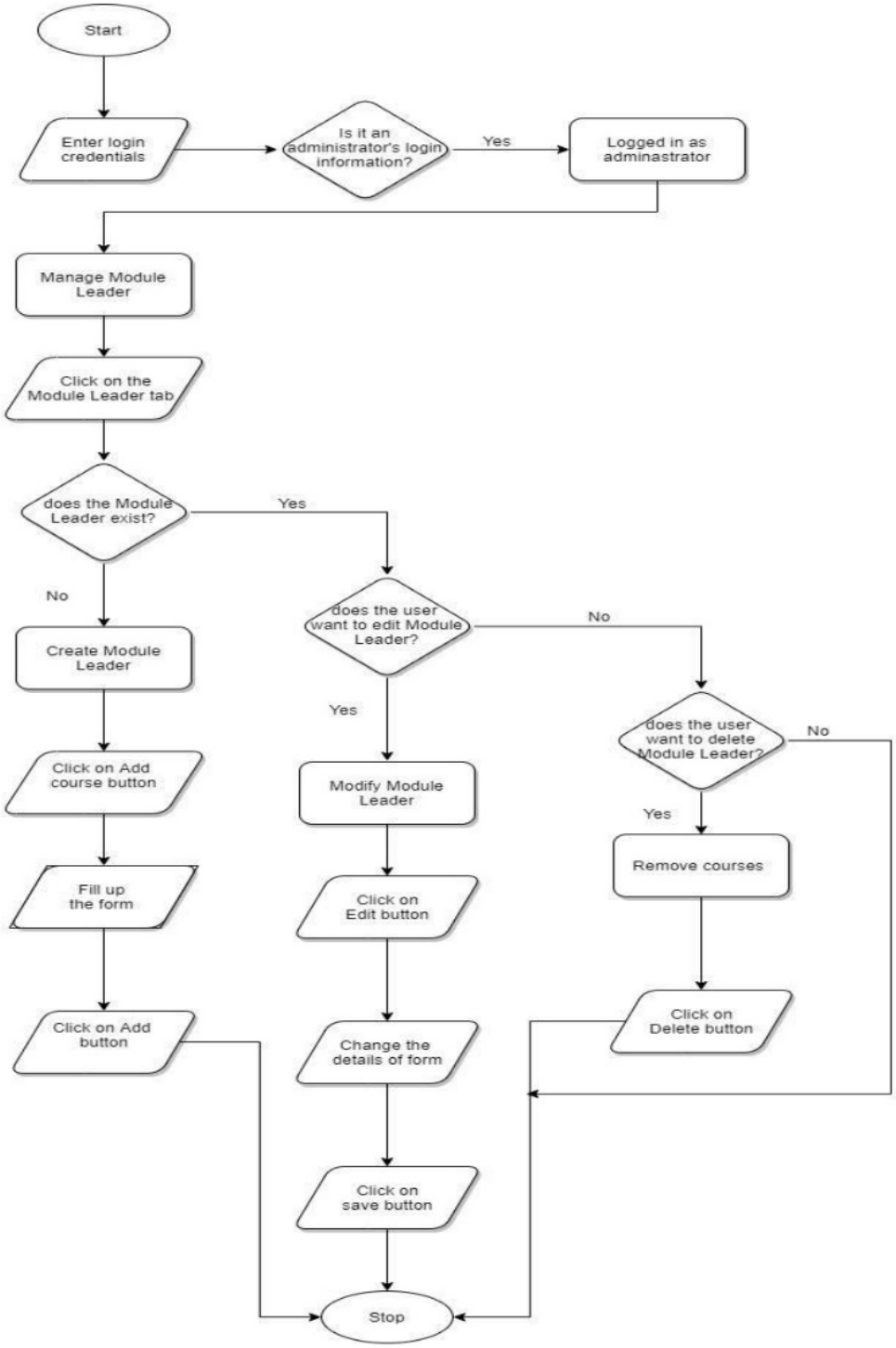


Figure 1. 28. Flowchart showing the process of modifying the module leader

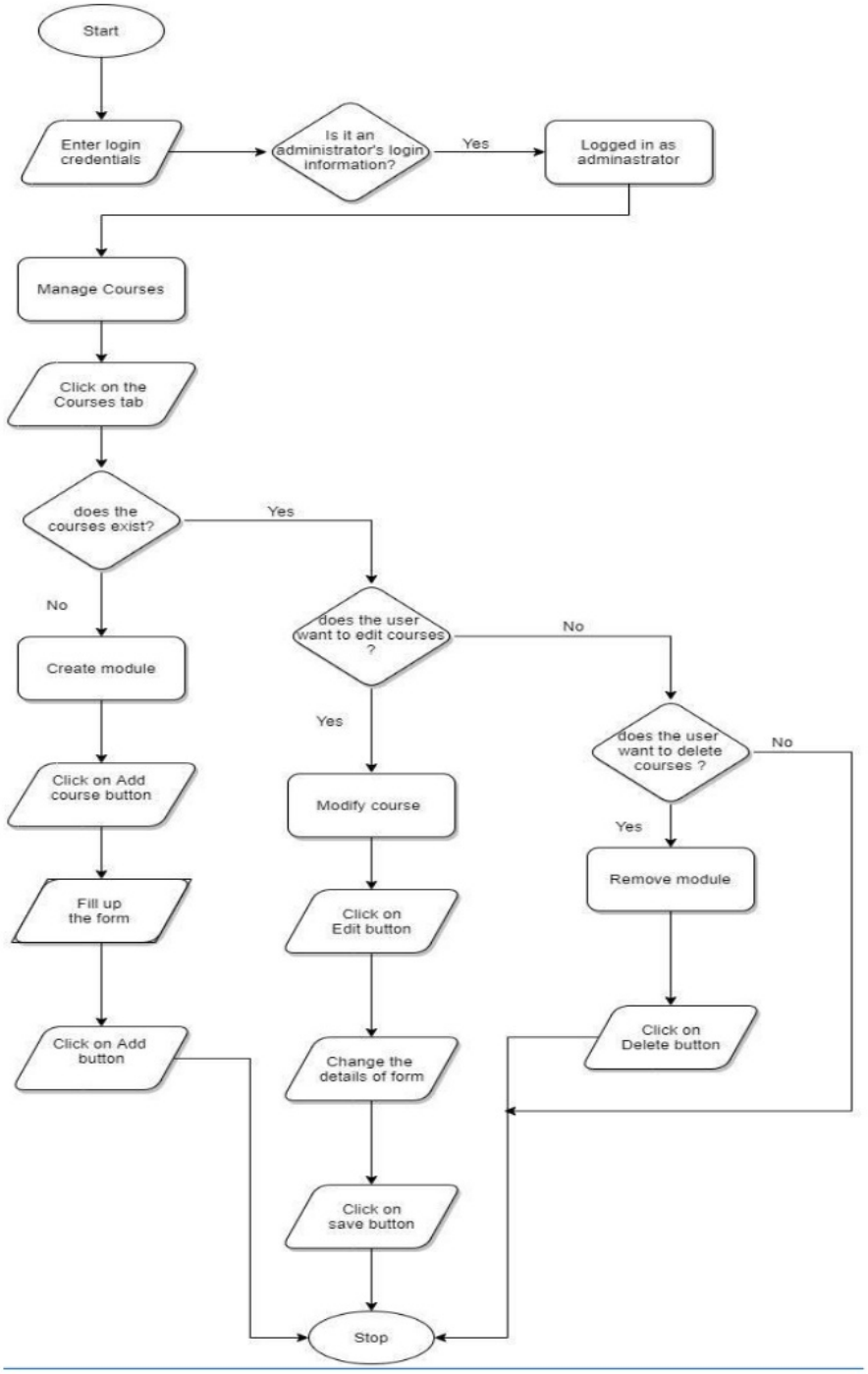


Figure 1. 29. Flowchart showing the process of modifying the course

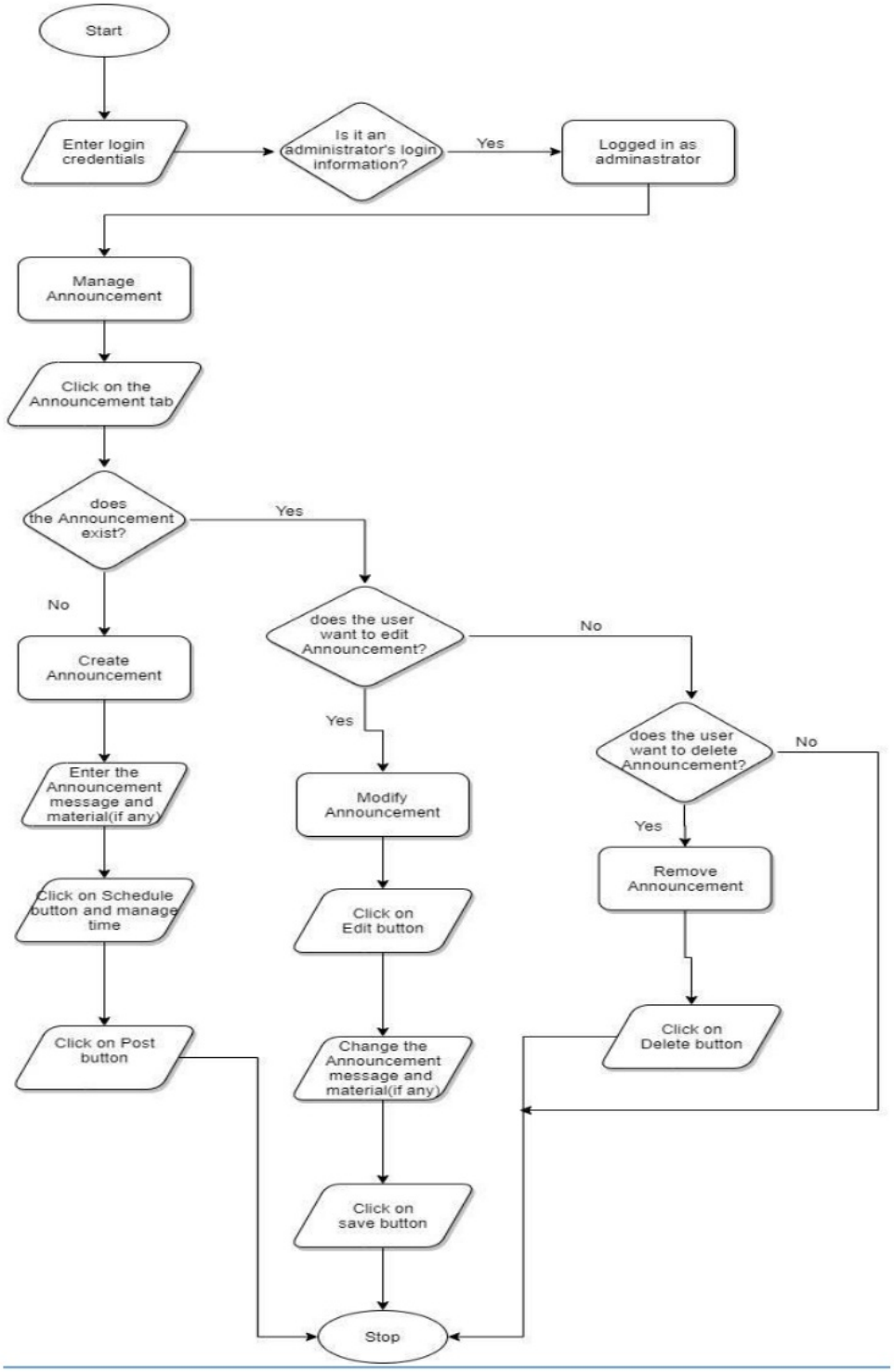


Figure 1. 30. Flowchart showing the process of adding announcements

5. System Build and Technical Notes

5.1 Technically Difficult Code Sections

1) Uploading UCAS file in the system

```
public function file()
{
    if(isset($_FILES['upload-file']) ){
        if(!empty($_FILES['upload-file']['name']) ){
            $fileName = $_FILES['upload-file']['name'];
            $fileTmp = $_FILES['upload-file']['tmp_name'];
            $validate_ext = array('xml','xlsx');
            $fileData = explode('.', $fileName);
            $fileExt = end($fileData);
            if(in_array($fileExt, $validate_ext)){
                $this->load->model('Database', 'import');
                $this->load->library('excel');
                $inputFileName = $_FILES['upload-file']['tmp_name'];
                $inputFileType = PHPExcel_IOFactory::identify($inputFileName);
                $objReader = PHPExcel_IOFactory::createReader($inputFileType);
                $objPHPExcel = $objReader->load($inputFileName);
                $allDataInSheet = $objPHPExcel->getActiveSheet()->toArray(null,true,true,true);
                $arrayCount = count($allDataInSheet);

                foreach ($allDataInSheet as $dataInSheet) {
                    foreach ($dataInSheet as $key => $value) {
                        $value = preg_replace('/\s+/', '', $value);
                        $SheetDataKey[trim($value)] = $key;
                        $a =  $SheetDataKey[trim($value)];
                    }
                }
                for ($i = 2; $i <= $arrayCount; $i++) {
                    $uid = $allDataInSheet[$i]["A"];
                    $ufname = $allDataInSheet[$i]["B"];
                    $umname = $allDataInSheet[$i]["C"];
                    $ulname = $allDataInSheet[$i]["D"];
                    $birthdate = $allDataInSheet[$i]["E"];
                    $address = $allDataInSheet[$i]["F"];
                    $gender = $allDataInSheet[$i]["G"];
```

Functions:

- Converting student ucase file to system read able formats
- Auto creating student password using Last name, date of birth etc.
- Dynamic insert file in the system

2) Student Attendance

```php
public function student_attendance(){
    $this->load->model('Database');
    if (isset($_POST['action']) && !empty($_POST['action']) && $_POST['action']=="Add") {
        if (empty($_POST['attendance_date'])) {
            $output = array(
                'error'=>true,
                'error_attendance_date'=>"Attendance Date is required",
            );
        }
        else{
            $check = $this->Database->checkAttendance($_POST['module_id'],$_POST['attendance_date'])->result();
            if(!isset($_POST['student_id'])){
                $output = array(
                    'error'=>true,
                    'error_attendance_date'=>"No Student record found.Please contact to system admin for more info.",
                );
                echo json_encode($output);
                exit();
            }
            if(empty($check) && isset($_POST['student_id'])){
                for($count = 0; $count < count($_POST['student_id']); $count++){
                    $data = array(
                        'student_id'=>$_POST['student_id'][$count],
                        'attendance_status'=>$_POST["attendance_status".$_POST['student_id'][$count]."]",
                        'date'=>    $_POST['attendance_date'],
                        'module_id'=>$_POST['module_id']
                    );
                    $fetchData[] = $data;
                    if($this->Database->insertAttendance($data)==200){
                        $output = array(
                            'success'=>'Data Added Successfully',
                        );
                    }
                    else{
                        $output = array(
                            'success'=>'Error',
                        );
                    }
                }
            }
            else{
                $output = array(
                    'error'=>true,
                    'error_attendance_date' =>  'Attendance Data Already Exists on this date'
                );
            }
        }
        echo json_encode($output);
    }
```

Functions:

- Taking attendance of system for respective module in respective dates
- Auto mail to the students having less attendance
- Track student attendance records

3. Student enroll confirmation e-mail

```php
private function confrimationEmail($id = "", $status= false){
// defined('valid_email') OR exit(show_404());
    $this->load->model('Database');
    $data=$this->Database->fetch_users($id);
    if($id=="" ||$status == false || $data==""){
        exit(show_404());
    }
    $this->load->library('email');
    $config['protocol']    = 'smtp';
    $config['smtp_host'] = 'ssl://smtp.gmail.com';
    $config['smtp_port'] = '465';
    $config['smtp_timeout'] = '700';
    $config['smtp_user'] = 'universitywoodland@gmail.com';
    $config['smtp_pass'] = 'GroupProoject@@@';
    $config['charset'] = 'utf-8';
    $config['newline'] = "\r\n";
    $config['mailtype'] = 'html';
    $config['validation'] = TRUE;

    $this->email->initialize($config);

    $message = '<!DOCTYPE html>
    <html>
    <head>
    <title>Tilte</title>
    <style type="text/css">
    body{
        width: 75%;
        margin: auto;
    }
    header{
        display: flex!important;
        background: #f8f9fa!important;
        margin: auto;
    }
    section{
        width: 75%;
        margin: 4% auto;
        font-family: sans-serif;
        font-size: 1rem;
        font-weight: 400;
        line-height: 1.5;
        color: #212529;
        text-align: left;
        background-color: #fff;
    }
    .navbar{
```

Functions:

- Sending auto confirmation email to the respective students.

5.2 On-going Testing Methodology

Testing plays a huge role in determining the success of a project. A proper testing structure leads to determining the flaws and errors of a system which eventually helps in making the quality of any project better. The testing methodology used hereby is black-box testing. In this method, a test case is considered and the end result is checked and compared to the actual result that is expected.

5.3 Final System Interface Displays

The final interface displays

5.3.1. Login page

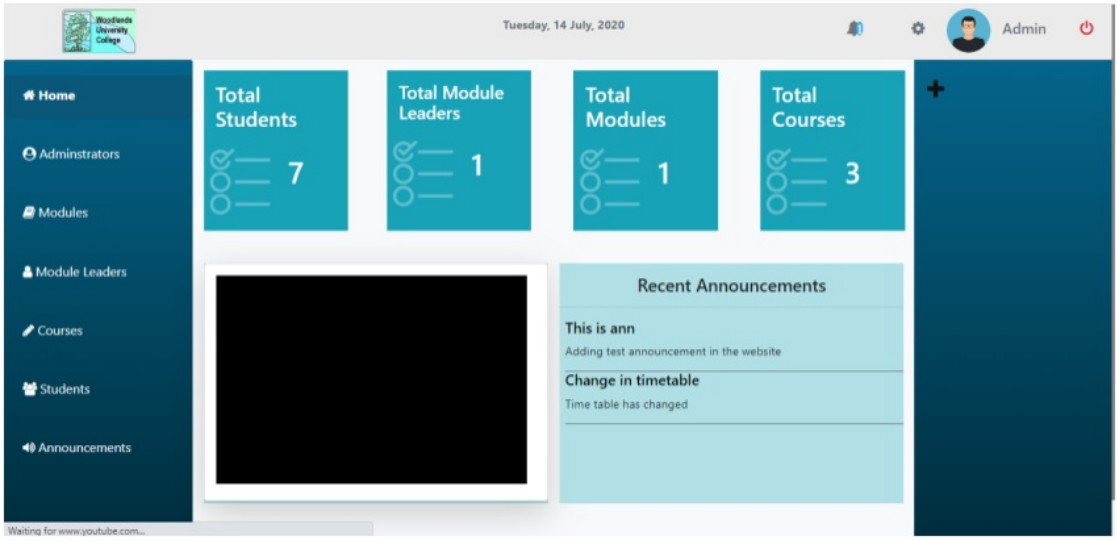


Figure 1. 31Admin Dashboard

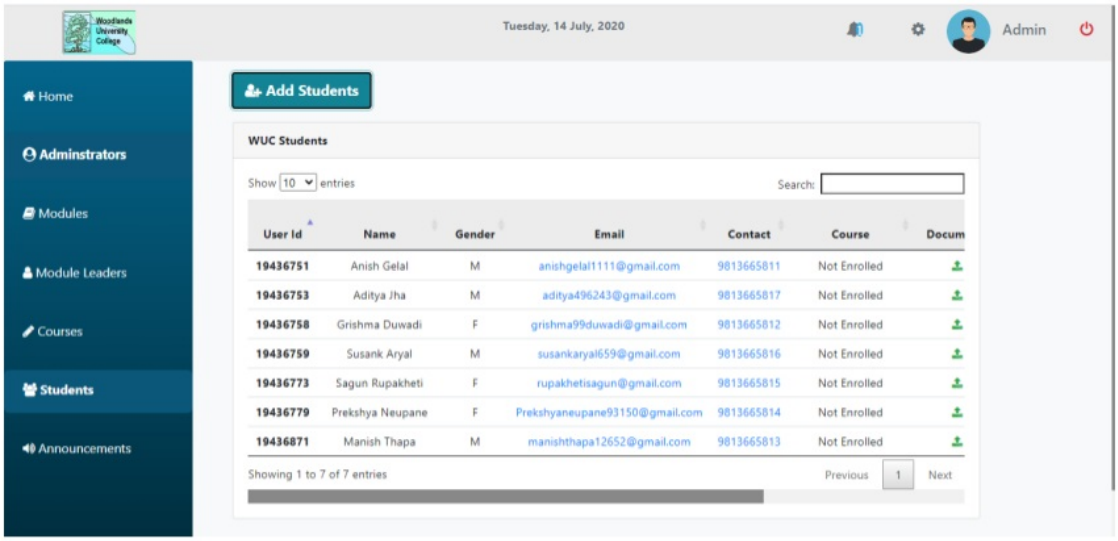


Figure 1. 32Admission process

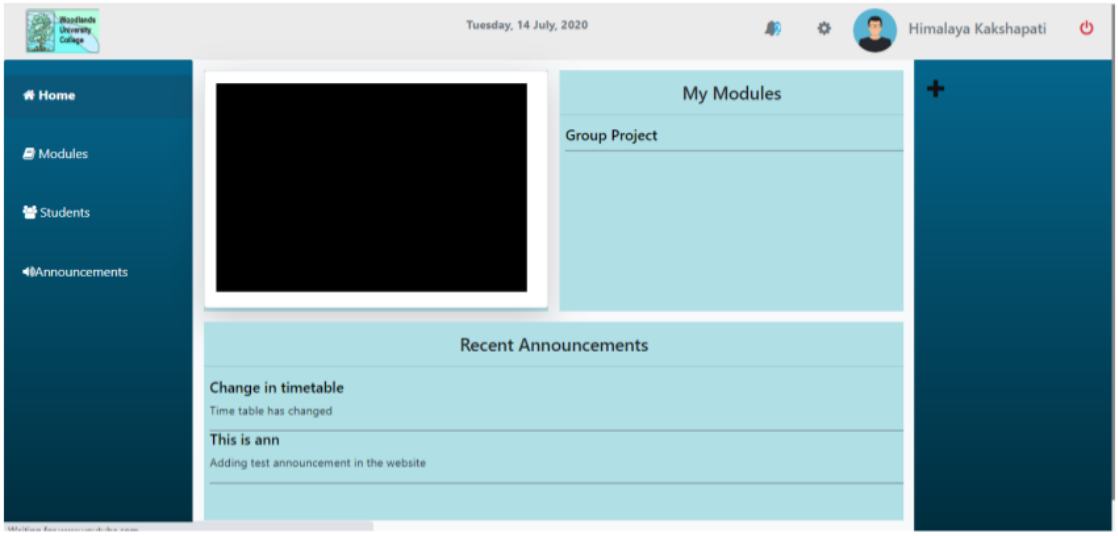


Figure 1. 33Module leader dashboard

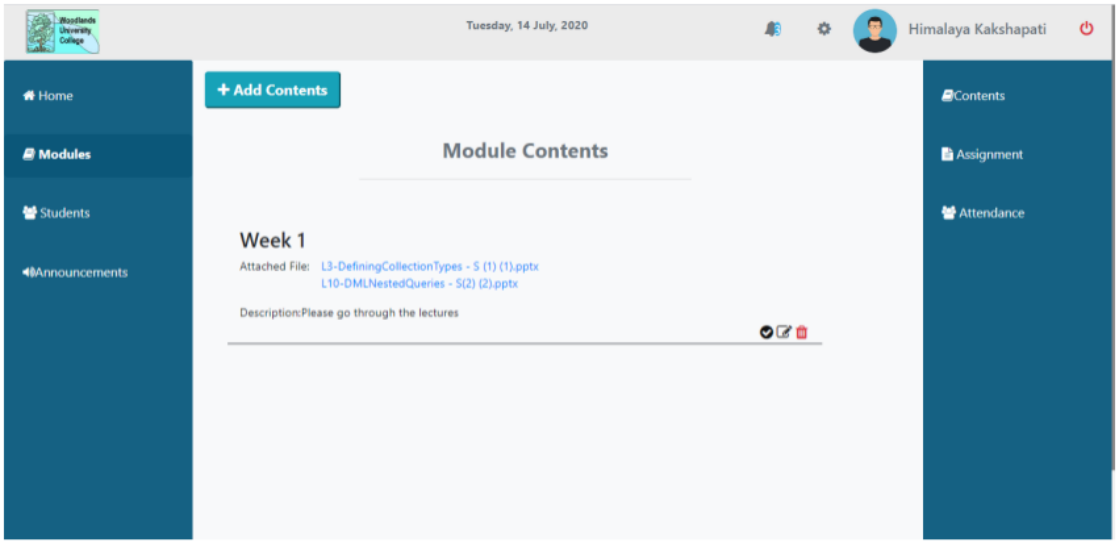


Figure 1. 34Module view

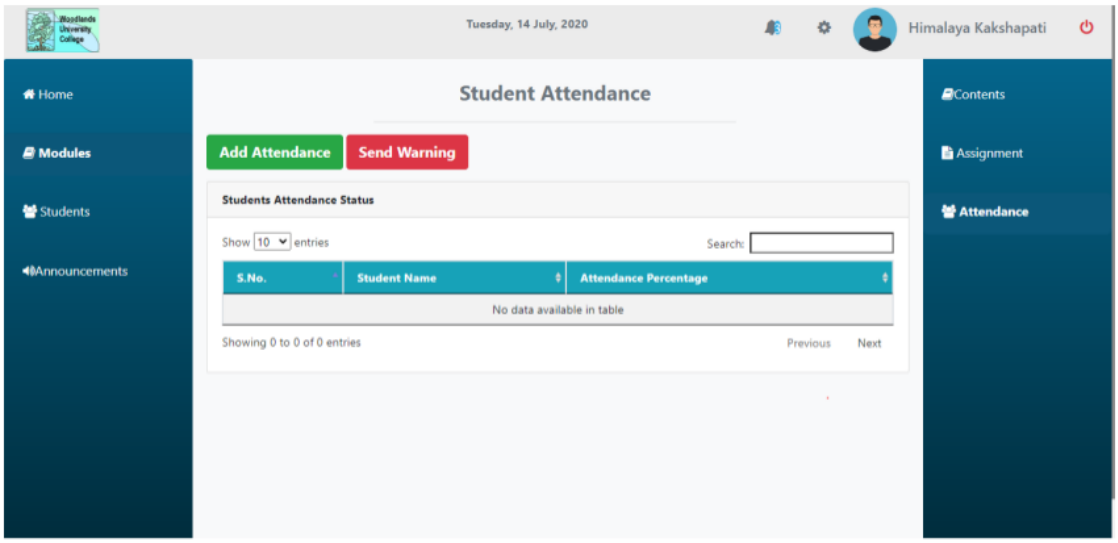


Figure 1. 35 Attendance view

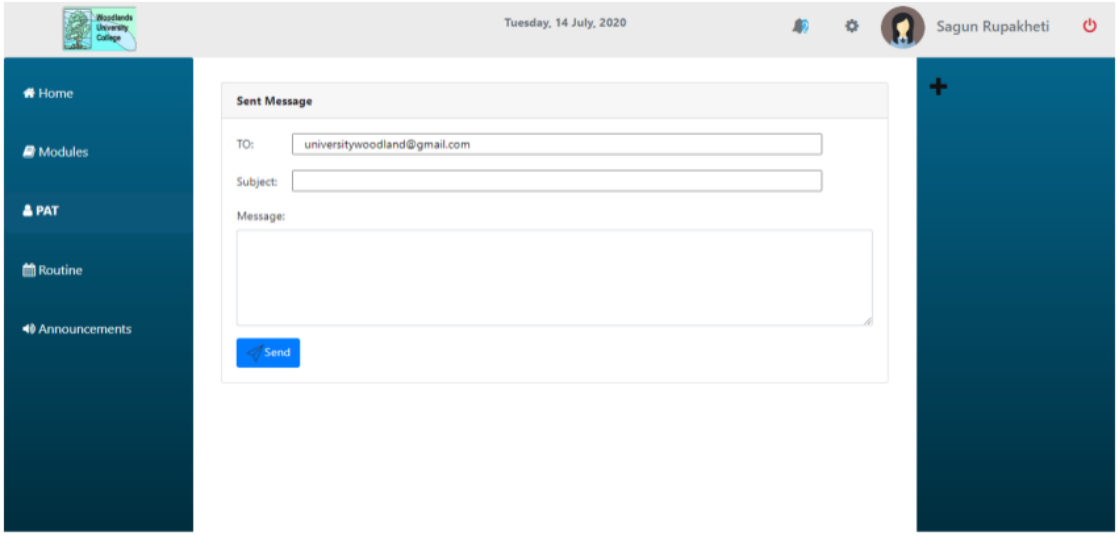


Figure 1. 36 PAT communication

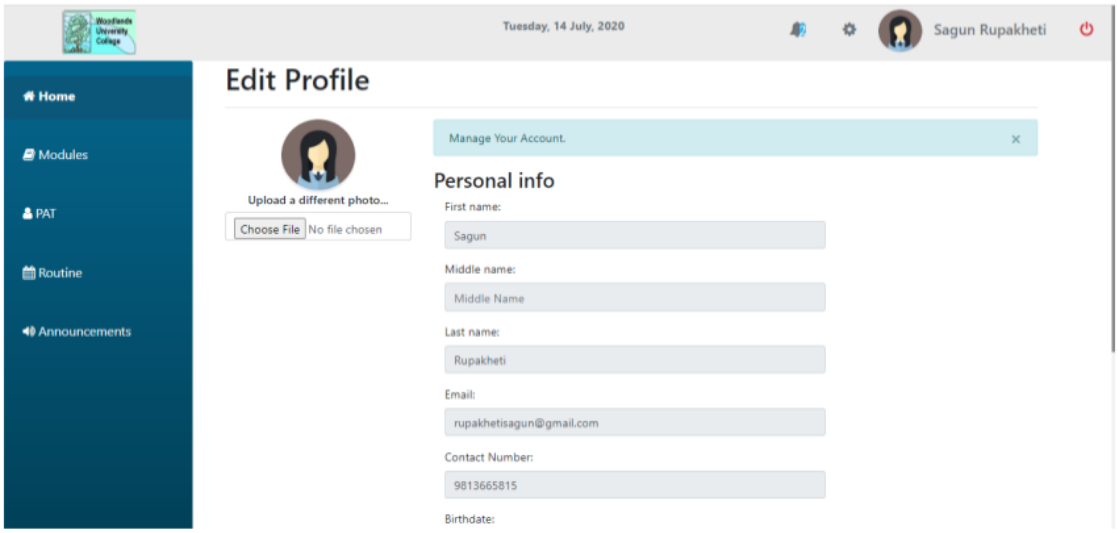


Figure 1. 37 Profile view

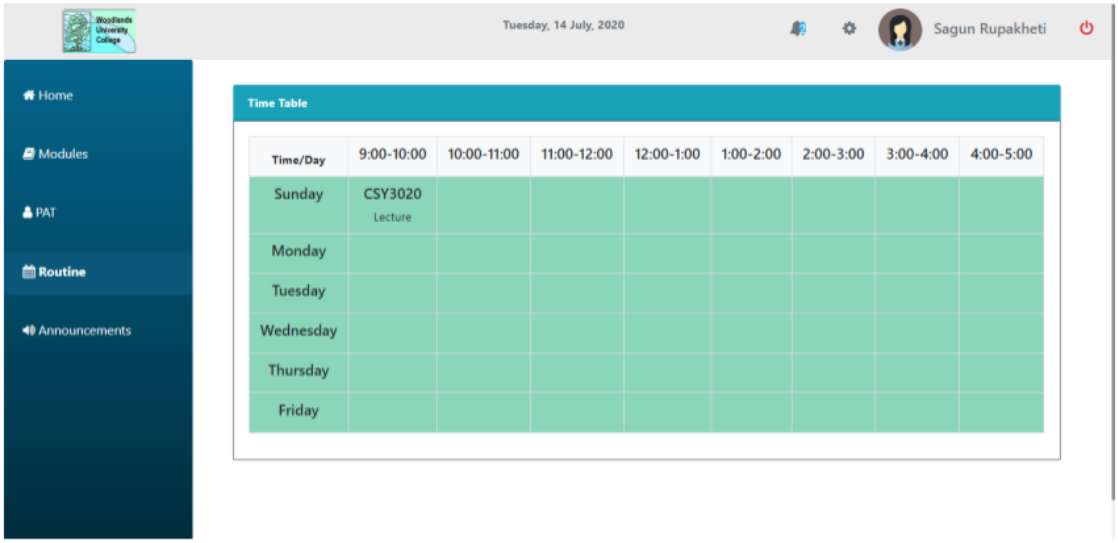


Figure 1. 38 Time-table view

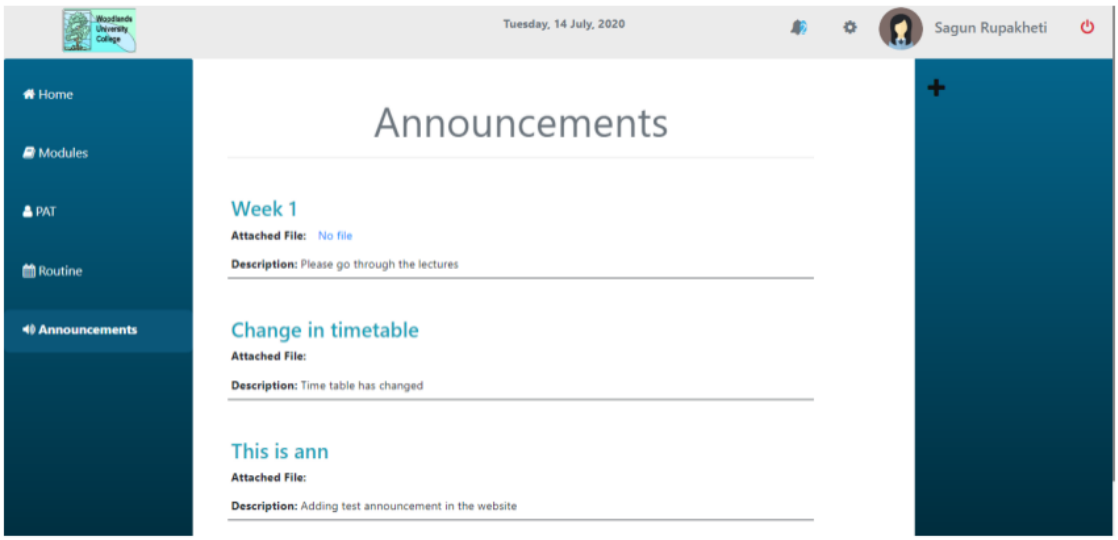


Figure 1. 39 Announcements view

6. Test Strategy

6.1 Overview of Test Strategy

The testing is done based on user expectation rather than code testing. Test cases are considered and expected and actual outputs are compared which shows whether a functionality works successfully or not.

6.2 Sample Test Results

Below are all the possible test cases for the system.

6.2.1. Administrator Testing

S.No.	Test Case	Process	Expected Output	Actual Output	Result
1. Admin Management					
1.	Add admin with all input fields filled	▪ Click on administrator tab ▪ Click on add admin ▪ Fill in the form and submit	Administrator added to the system	Administrator added to the system	Successful ✓
2.	Add admin with all some/none input fields filled	▪ Click on administrator tab ▪ Click on add admin ▪ Fill in the form partially and submit	Warning message shown to fill in all the fields	Warning message shown to fill in all the fields	Successful ✓

3.	Edit admin information with the form filled completely	▪ Click on administrator tab ▪ Click on edit on the record to be edited ▪ Edit the information and submit	Record successfully edited	Record successfully edited	Successful ✓
4.	Edit admin information with the form filled partially	▪ Click on administrator tab ▪ Click on edit on the record to be edited ▪ Edit the information and submit	Warning message shown to fill in all the fields	Warning message shown to fill in all the fields	Successful ✓
5.	Delete admin information	▪ Click on administrator tab ▪ Click on delete on the record to be deleted ▪ Confirm the action	Record deleted	Record deleted	Successful ✓
2. Module Leaders Management					

6.	Add module leader with form filled completely	▪ Click on Module Leaders tab ▪ Click on Add Module Leader ▪ Fill the form and submit	Record added	Record added	Successful ✓
6.	Add module leader with partially filled form	▪ Click on Module Leaders tab ▪ Click on Add Module Leader ▪ Fill the form and submit	Warning message shown to fill in all the fields	Warning message shown to fill in all the fields	Successful ✓
7.	Edit module leader information with completely filled form	▪ Click on Module Leaders tab	Record successfully edited	Record successfully edited	Successful ✓

		▪ Click on edit on the particular record to edit ▪ Edit the information and submit			
8.	Edit module leader information with partially filled form	▪ Click on Module Leaders tab ▪ Click on edit on the particular record to edit ▪ Edit the information and submit	Warning message shown to fill in all the fields	Warning message shown to fill in all the fields	Successful ✓
9.	Delete module leader information	▪ Click on Module Leaders tab	Record deleted	Record deleted	Successful ✓

		▪ Click on delete on the particular record ▪ Confirm the action			
10.	Change status of module leader	▪ Click on Module Leaders tab ▪ Click on status of the particular record ▪ Choose the status and submit	Status changed	Status changed	Successful ✓
3. Admission management/ Student records Management					
11.	Add student with completely filled form	▪ Click on Students tab ▪ Click on Add Students	Record /records added	Record/records added	Successful ✓

		▪ Choose a excel spreadsheet that contains the UCAS sent information ▪ Submit the file			
12.	Add student with valid excel spreadsheet	▪ Click on Students tab ▪ Click on Add Students ▪ Choose a excel spreadsheet that contains the UCAS sent information ▪ Submit the file	Warning message shown to fill in all the fields	Warning message shown to fill in all the fields	Successful ✓
13.	Edit student record with invalid excel spreadsheet	▪ Click on Students tab ▪ Click on edit on the particular record to edit ▪ Edit the information and submit	Record successfully edited	Record successfully edited	Successful ✓

14.	Edit student record with partially filled form	▪ Click on Students tab ▪ Click on edit on the particular record to edit ▪ Edit the information and submit	Warning message shown to fill in all the fields	Warning message shown to fill in all the fields	Successful ✓
15.	Delete student record	▪ Click on Students tab ▪ Click on delete on the particular record ▪ Confirm the action	Record deleted	Record deleted	Successful ✓

16.	Change student status	▪ Click on Students tab ▪ Click on status of the particular record ▪ Choose the status and submit	Status changed	Status changed	Successful ✓
17.	Add student documents with valid file/document	▪ Click on Students tab ▪ Click on documents on a particular record ▪ Choose documents and submit	Files added successfully	Files added successfully	Successful ✓
18.	Add student documents with invalid file/document	▪ Click on Students tab ▪ Click on documents on a particular record	Error message shown	Error message shown	Successful ✓

		▪ Choose documents and submit			
4. Course Management					
19.	Add course with completely filled form	▪ Click on Courses tab ▪ Click on Add Courses ▪ Fill in the form and submit	Course added	Course added	Successful ✓
20.	Add course with partially filled form	▪ Click on Courses tab ▪ Click on Add Courses ▪ Fill in the form and submit	Error message shown as to fill the form completely	Error message shown as to fill the form completely	Successful ✓

21.	Edit course with completely filled form	▪ Click on Courses tab ▪ Click on particular record's edit to edit the information ▪ Edit the information in the form and submit	Course successfully edited	Course successfully edited	Successful ✓
22.	Edit course with partially filled form	▪ Click on Courses tab ▪ Click on particular record's edit to edit the information ▪ Edit the information in the form and submit	Error message shown as to fill the form completely	Error message shown as to fill the form completely	Successful ✓
23.	Delete course	▪ Click on Courses tab ▪ Click on delete of the record to be deleted	Course deleted	Course deleted	Successful ✓

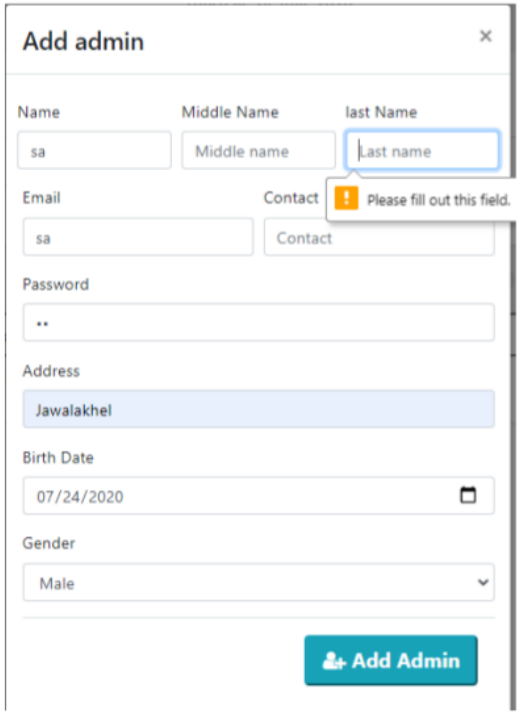
		▪ Confirm the action			
5. Module Management					
24.	Add module with completely filled form	▪ Click on Modules tab ▪ Click on Add Modules ▪ Fill in the form and submit	Module added	Module added	Successful ✓
25.	Add module with partially filled form	▪ Click on Modules tab ▪ Click on Add Modules ▪ Fill in the form and submit	Error message shown as to fill the form completely	Error message shown as to fill the form completely	Successful ✓

26.	Edit module with completely filled form	▪ Click on Modules tab ▪ Click on particular record's edit to edit the information ▪ Edit the information in the form and submit	Module successfully edited	Module successfully edited	Successful ✓
27.	Edit module with partially filled form	▪ Click on Modules tab ▪ Click on particular record's edit to edit the information ▪ Edit the information in the form and submit	Error message shown as to fill the form completely	Error message shown as to fill the form completely	Successful ✓
28.	Delete module	▪ Click on Modules tab ▪ Click on delete of the record to be deleted	Module deleted	Module deleted	Successful ✓

		▪ Confirm the action			
6. Announcement Management					
29.	Add announcement with completely filled form	▪ Click on Announcements tab ▪ Click on Add Announcements ▪ Fill in the form and submit	Announcement added	Announcement added	Successful ✓
29.	Add announcement with completely filled form	▪ Click on Announcements tab ▪ Click on Add Announcements ▪ Fill in the form and submit	Error message shown as to fill the form completely	Error message shown as to fill the form completely	Successful ✓

30.	Edit announcement with completely filled form	▪ Click on Announcements tab ▪ Click on particular record's edit to edit the information ▪ Edit the information in the form and submit	Announcement successfully edited	Announcement successfully edited	Successful ✓
31.	Edit announcement with partially filled form	▪ Click on Announcements tab ▪ Click on particular record's edit to edit the information ▪ Edit the information in the form and submit	Error message shown as to fill the form completely	Error message shown as to fill the form completely	Successful ✓
32.	Delete announcement	▪ Click on Announcements tab ▪ Click on delete of the record to be deleted	Announcement deleted	Announcement deleted	Successful ✓

		▪ Confirm the action			
7.Time-table Management					
33.	Add timetable	▪ Click on timetable in course tab ▪ Fill in the timetable	Time-Table added	Time-table added	Successful ✓
34.	Edit timetable	▪ Click on timetable in course tab ▪ Fill in the cell	Time-table edited	Time-table edited	Successful ✓



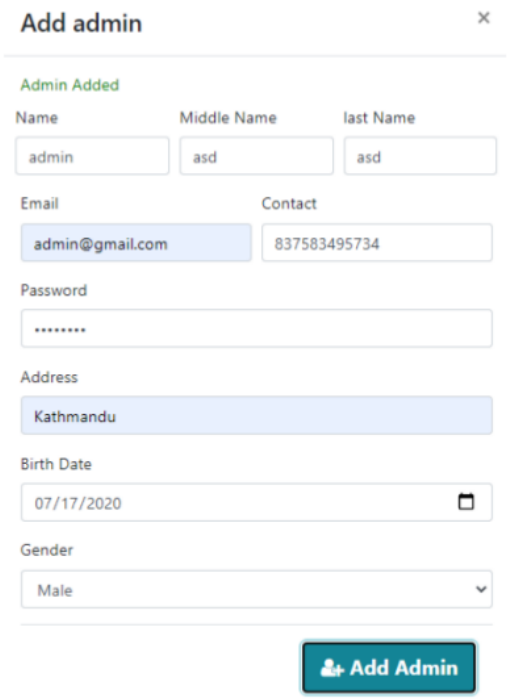


Figure 1. 40 Admin insertion successful and failure

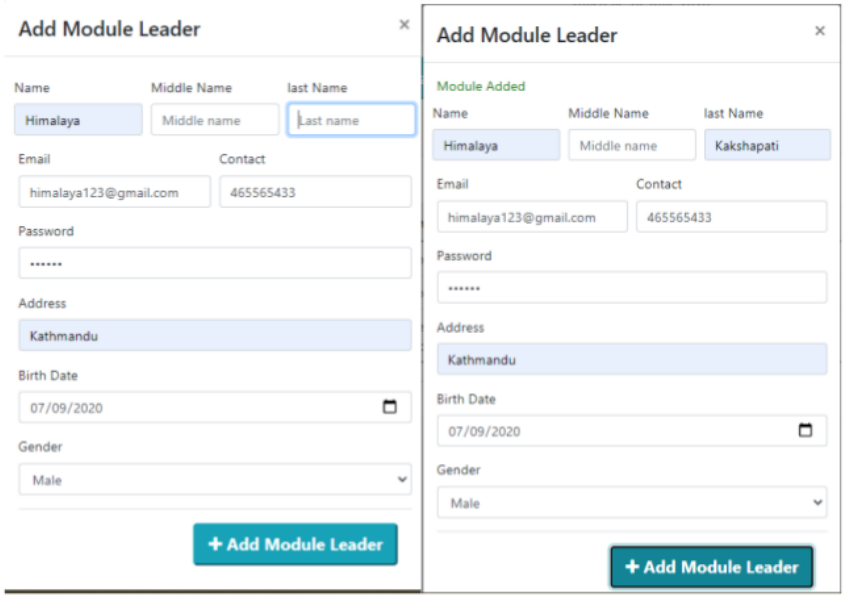


Figure 1.41 Module Leader insertion success and failure

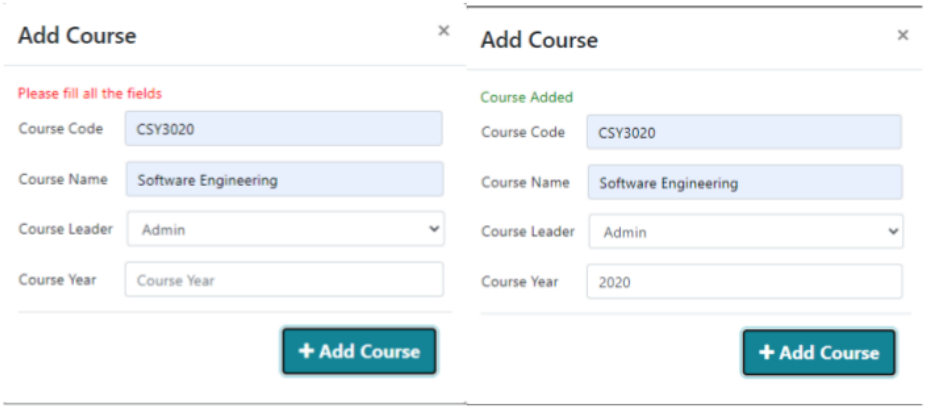
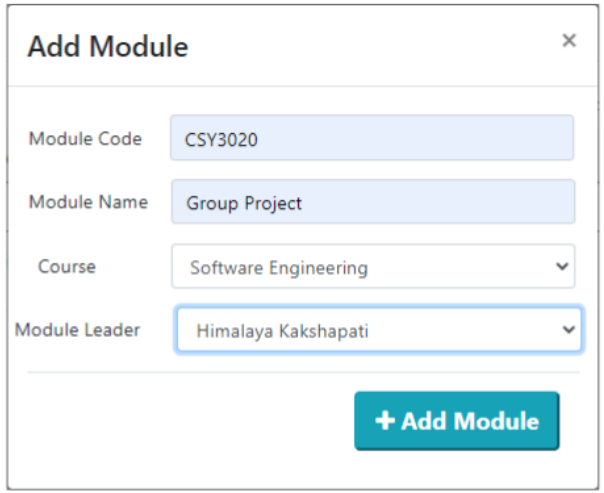


Figure 1. 42. Course insertion success and failure



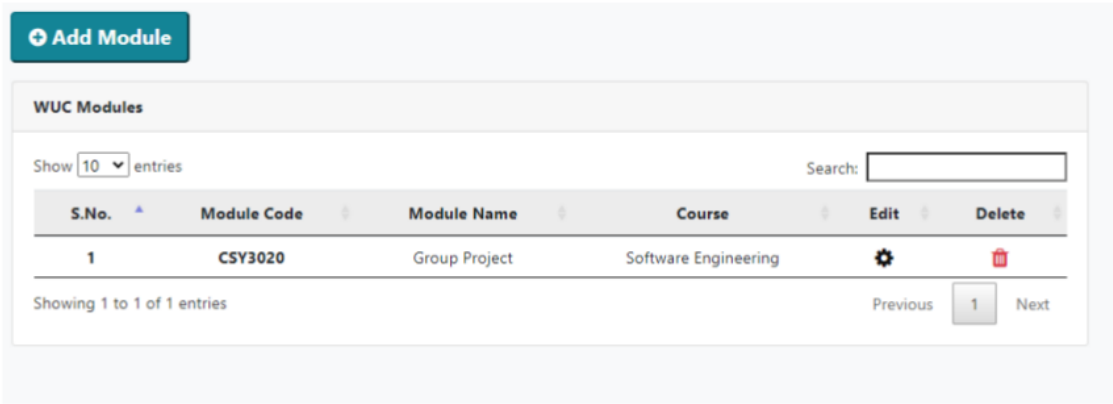


Figure 1. 43 Module insertion success

6.2.2. Module Leaders Testing

S.No.	Test Case	Process	Expected Output	Actual Output	Result
1. Modules Content Management					
1.	Add Content with all form inputs filled	▪ Click on view in modules tab ▪ Click on contents tab ▪ Click on Add Contents button ▪ Choose files, fill form and submit	Files added to the module	Files added to the module	Successful ✓
2.	Add Content with all partially filled form	▪ Click on view in modules tab ▪ Click on contents tab ▪ Click on Add Contents button ▪ Choose files, fill form and submit	Error shown	Error shown	Successful ✓
3.	Edit Content with completely filled form data	▪ Click on view in modules tab ▪ Click on contents tab ▪ Click on edit	Content information edited	Content information edited	Successful ✓

		▪ Edit the information and submit			
4.	Edit Content with partially filled form data	▪ Click on view in modules tab ▪ Click on contents tab ▪ Click on edit ▪ Edit the information and submit	Error shown	Error shown	Successful ✓
5.	Delete Content	▪ Click on view in modules tab ▪ Click on contents tab ▪ Click on delete ▪ Confirm the action	Content deleted	Content deleted	Successful ✓
6.	Change visibility	▪ Click on view in modules tab ▪ Click on contents tab ▪ Click on visibility	Visibility changed	Visibility changed	Successful ✓
2. Assignment Management					

7.	Add Assignment with all inputs filled	▪ Click on view in modules tab ▪ Click on contents tab ▪ Click on Add Assignment button ▪ Choose files, fill form and submit	Assignment added to the module	Assignment added to the module	Successful ✓
8.	Add Assignment with some/none inputs filled	▪ Click on view in modules tab ▪ Click on contents tab ▪ Click on Add Assignment button ▪ Choose files, fill form and submit	Error shown	Error shown	Successful ✓
9.	Edit Assignment with all inputs filled	▪ Click on view in modules tab ▪ Click on contents tab ▪ Click on Edit button ▪ Choose files, fill form and submit	Assignment information edited	Assignment information edited	Successful ✓
10.	Edit Assignment with some/none inputs filled	▪ Click on view in modules tab ▪ Click on contents tab ▪ Click on Edit button	Error shown	Error shown	Successful ✓

		▪ Choose files, fill form and submit			
11.	Delete Assignment	▪ Click on view in modules tab ▪ Click on Assignment tab ▪ Click on delete button ▪ Confirm action	Assignment deleted	Assignment deleted	Successful ✓
12.	Change visibility	▪ Click on view in modules tab ▪ Click on Assignment tab ▪ Click on visibility	Visibility changed	Visibility changed	Successful ✓
3. Grade Management					
13.	Add grade	▪ Click on view in modules tab ▪ Click on Assignment tab ▪ Click on grades ▪ Choose the grade for student and add feedback and submit	Grade added	Grade added	Successful ✓

14.	Edit grade	▪ Click on view in modules tab ▪ Click on Assignment tab ▪ Click on grades ▪ Click on the grade to edit ▪ Edit the grade and submit	Grade edited	Grade edited	Successful ✓
15.	Delete grade	▪ Click on view in modules tab ▪ Click on Assignment tab ▪ Click on grades ▪ Click on the grade to delete ▪ Delete the student's grade	Grade deleted	Grade deleted	Successful ✓
4. Attendance Management					
16.	Add attendance	▪ Click on view in modules tab ▪ Click on Attendance tab ▪ Click on Add Attendance	Attendance added	Attendance added	Successful ✓

		▪ Select the present students and click on submit			
17.	Send warning for low attendance	▪ Click on view in modules tab ▪ Choose the module and click on view ▪ Click on Attendance tab ▪ Click on Send Warning button ▪ Choose the number of absent days and click submit	Warning sent to students having low attendance	Warning sent to students having low attendance	Successful ✓
5. PAT					
18.	Send email to student for meeting	▪ Click on PAT tab ▪ Click on the student to send the email ▪ Fill the form and send.	Email sent for meeting schedule	Email sent for meeting schedule	Successful ✓

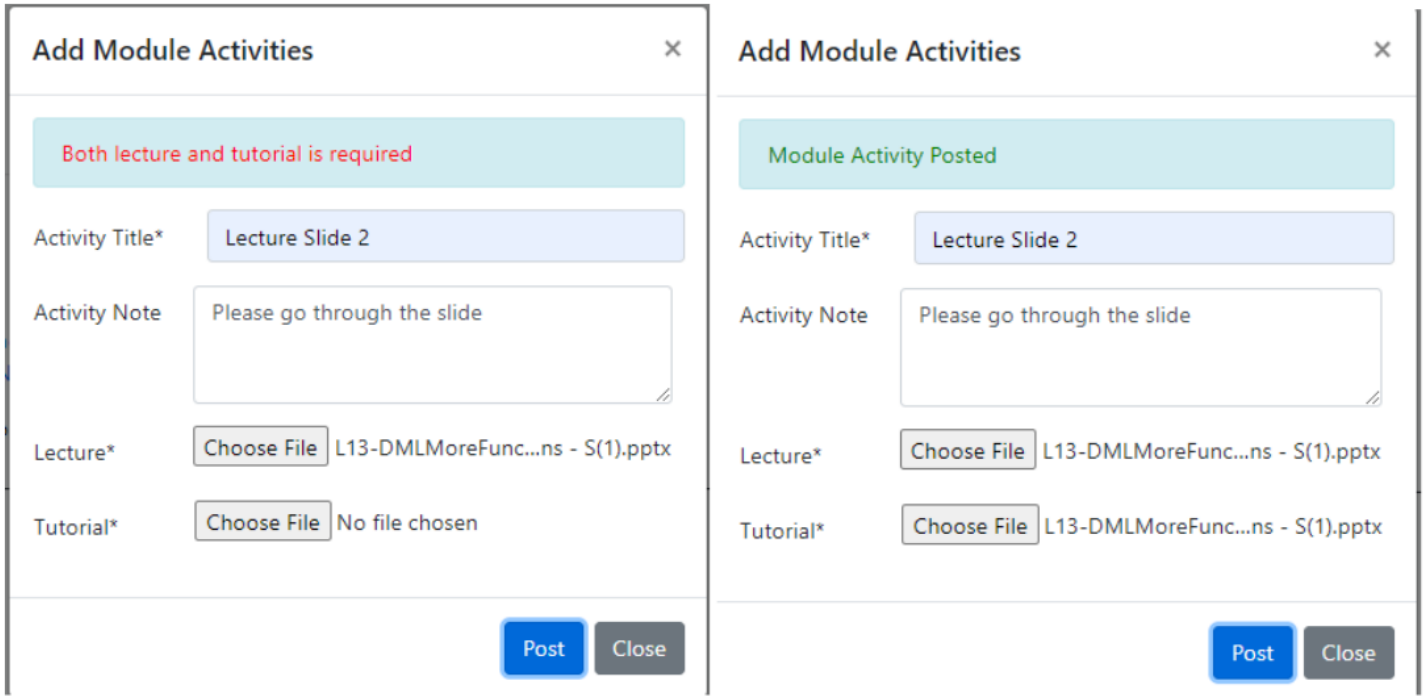


Figure 1. 44. Add Module activities success and failure

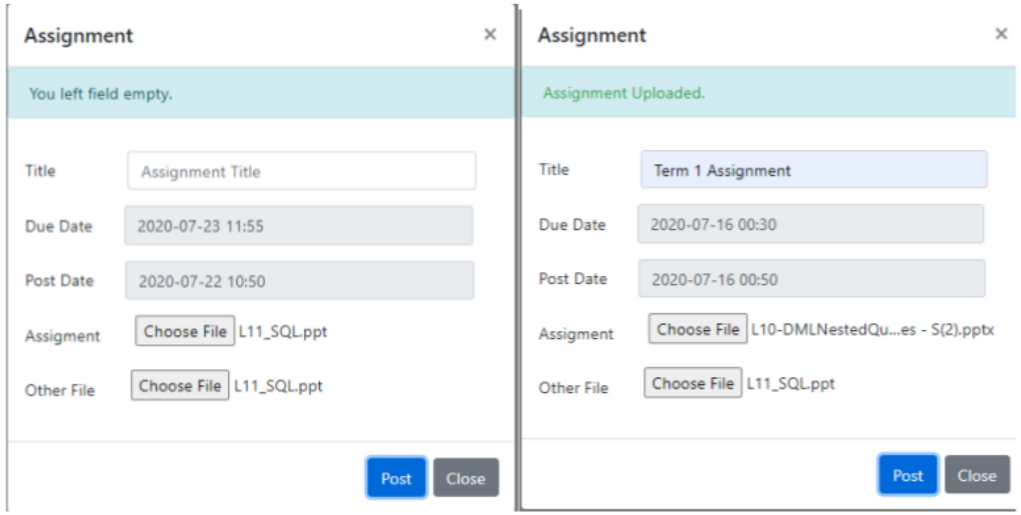


Figure 1. 45. Assignment upload success and failure

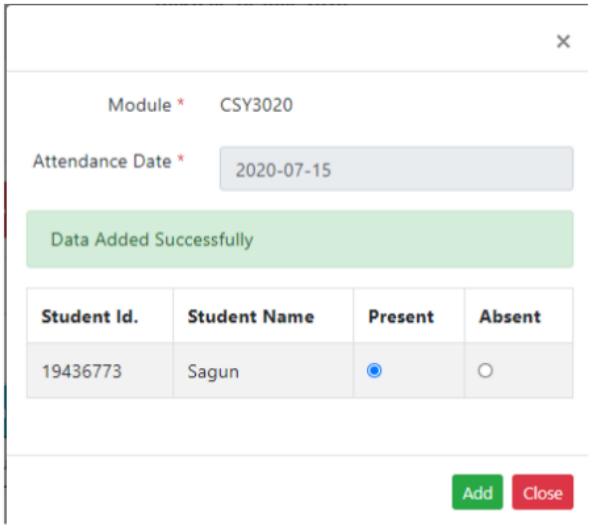


Figure 1. 46. Adding attendance

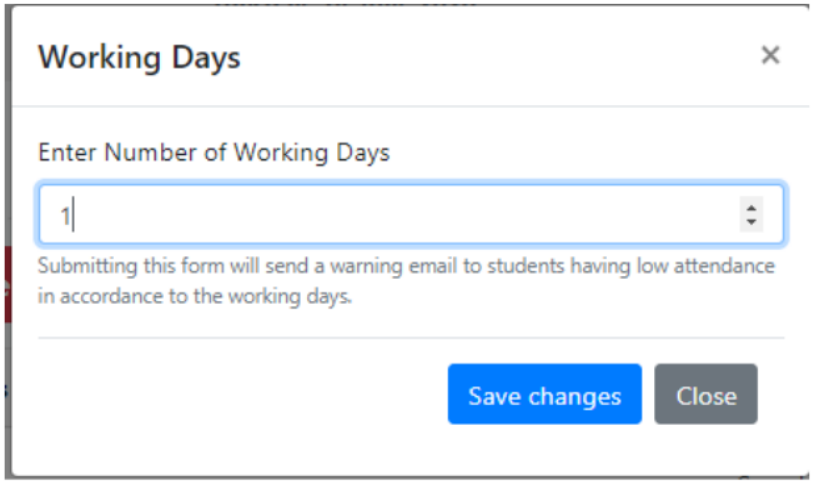


Figure 1.47. Low-attendance warning

6.2.3. Students Testing

S.No.	Test Case	Process	Expected Output	Actual Output	Result
1. Assignment submission					
1.	Submit assignment within deadline	▪ Click on view in modules tab ▪ Choose the module and click on view ▪ Click on the Assignment tab ▪ Click on Submit Assignment button ▪ Choose files and submit	Assignment submitted with on time remarks	Assignment submitted with on time remarks	Successful ✓
2.	Submit assignment post deadline	▪ Click on view in modules tab ▪ Choose the module and click on view ▪ Click on the Assignment tab	Assignment submitted with late submission remarks	Assignment submitted with late submission remarks	Successful ✓

		▪ Click on Submit Assignment button ▪ Choose files and submit			
2. PAT					
3.	Send email to PAT for meeting	▪ Click on PAT tab ▪ Fill the form and send.	Email sent for meeting schedule	Email sent for meeting schedule	Successful ✓
3. Modules					
4.	View module contents	▪ Click on Modules tab ▪ Click on view	Can view module contents, assignments and grades	Can view module contents, assignments and grades	Successful ✓
5.	Download module contents	▪ Click on Modules tab ▪ Click on view ▪ Click on contents ▪ Download the contents	Successfully downloaded	Successfully downloaded	Successful ✓

6.2.4. All user groups

S.No.	Test Case	Process	Expected Output	Actual Output	Result
1. Login					
1.	Login with correct login credentials	▪ Fill in the form ▪ Submit	Directed to respective user dashboard	Directed to respective user dashboard	Successful ✓
1.	Login with correct wrong credentials	▪ Fill in the form ▪ Submit	Error message shown	Error message shown	Successful ✓
2. Logout					
3.	Logout of the system	▪ Click on logout button in the header	Logged out of the system	Logged out of the system	Successful ✓

Figure 1. 48 Wrong login credentials

7. System Evaluation

7.1 Usability Evaluation Strategy

Evaluation is the process to examine the flexibility and usability of a system. A series of questionnaire were prepared in order to ask a few people associated to WUC who would be asked to use the system and evaluate it based on their experiences. The user will be given 5 options to choose their experience from.

The questions to be asked to:

1. How was your experience with the system overall?
2. How was the system complexity?
3. How well could you perform desired functionalities?
4. Did the functionalities work as expected?
5. Do you think people would need guidance to use the system?
6. Were there any errors that you found in the system?
7. Do you think you will get used to using the system without any problem?

7.2 System Pilot Trials

Candidate 1:

1. How was your experience with the system overall?

Bad	
Could be worse	
Moderate	

Satisfactory	✓
Excellent	

2. How was the system complexity?

Bad	
Could be worse	
Moderate	✓
Satisfactory	
Excellent	

3. How well could you perform desired functionalities?

Bad	
Could be worse	
Moderate	
Satisfactory	
Excellent	✓

4. Did the functionalities work as expected?

Yes	✓

No	

5. Do you think people would need guidance to use the system?

Yes	✓
No	

6. Were there any errors that you found in the system?

Yes	✓
No	

7. Do you think you will get used to using the system without any problem?

Yes	✓
No	

Candidate 2:

1. How was your experience with the system overall?

Bad	
Could be worse	

Moderate	
Satisfactory	
Excellent	✓

2. How was the system complexity?

Bad	
Could be worse	
Moderate	
Satisfactory	✓
Excellent	

3. How well could you perform desired functionalities?

Bad	
Could be worse	
Moderate	
Satisfactory	
Excellent	✓

4. Did the functionalities work as expected?

Yes	✓
No	

5. Do you think people would need guidance to use the system?

Yes	✓
No	

6. Were there any errors that you found in the system?

Yes	
No	✓

7. Do you think you will get used to using the system without any problem?

Yes	✓
No	

Candidate 3:

1. How was your experience with the system overall?

Bad	

Could be worse	
Moderate	✓
Satisfactory	
Excellent	

2. How was the system complexity?

Bad	
Could be worse	
Moderate	
Satisfactory	✓
Excellent	

3. How well could you perform desired functionalities?

Bad	
Could be worse	
Moderate	✓
Satisfactory	
Excellent	

4. Did the functionalities work as expected?

Yes	✓
No	

5. Do you think people would need guidance to use the system?

Yes	✓
No	

6. Were there any errors that you found in the system?

Yes	
No	✓

7. Do you think you will get used to using the system without any problem?

Yes	✓
No	

7.3 System Trial Results

From the above system trials, the below conclusions have been drawn.

S.no.	System Evaluation	Average Evaluation Results
1.	System Usability	Easy to use
2.	System Complexity	Moderate
3.	System Functionalities	Work as expected
4.	Bugs and Errors	None
5.	Assistance	Required
6.	Overall experience	Satisfactory
7.	Learning	Moderate

8. Project Conclusions

Overall, the system works as a course management system for Woodlands University College (WUC). The basic requirements of adding, amending , deleting and viewing various functionalities is available extensively. The project has taken around 28 weeks (after interview) for completion.

The first and the most important user area is considered to be the administrator area which was briefed by Dr. Simon White and Mr. Adam Blake. The main areas that administrators cover are handling and adding various different sectors in the system. The core areas covered by the admin area are:

- **Administrator Management**
- **Module Leader Management**
- **Admission Process and Management**
- **Students Management**
- **Course Management**
- **Modules Management**
- **Announcement Management**

The second user area is module leader area which was briefed by Dr. Raj Singh who is also a senior lecturer and module leader in WUC. The core areas covered by the module leaders are:

- **Module Content Management**
- **Assignment Management**

- **Grade Management**
- **PAT Communication**

The last user group is the student area which was briefed by Mr. Mark Williams who is a current student enrolled in computing course. The functionalities for the students are:

- **Add Assignment**
- **View Grades, View Announcements**
- **PAT Communication**

The system is not a complete functional working system for real life use. It still needs amendments and addition of various functionalities in order to be used for real-world scenario. For the implementation/code codeigniter framework, bootstrap and MVC pattern have been used to make the code more efficient. Testing has been done extensively to check if all the functionalities work well and correctly.

The project mainly focuses on team works and working in various fields to gain the experience and knowledge about different documentation terminologies and logical functionalities. It has provided a better view and perspective for the team members for future real-time projects which solely requires working in teams and groups.

9. Project Management

The proper guidance and support throughout the project lifespan to achieve the main target and goal of the project is the most important trait of project management. The main goal is to achieve all the demands and goals within the given time frame. Below are different aspects that determine the proper management of a project.

9.1. Project Gantt Chart/WBDS/Activity List

Gantt chart is a type of bar diagram that gives the information about the project according to the time elapsed and schedules of various tasks of a project. It helps to determine the main elements that were focused in the system.

Below are 2 gantt charts that show the time elapsed for documentation and implementation respectively.

Documentation

Topics	Start Week	End Week	Total Weeks
Introduction	1	2	1
Requirements Engineering	2	6	4
System Interface Designs	6	12	6
System Analysis and Design	12	16	4
System Build and Notes	17	18	1
Test Strategy	18	21	3
System Evaluation	21	22	1
Project Conclusions	22	23	1
Project Management	23	26	3

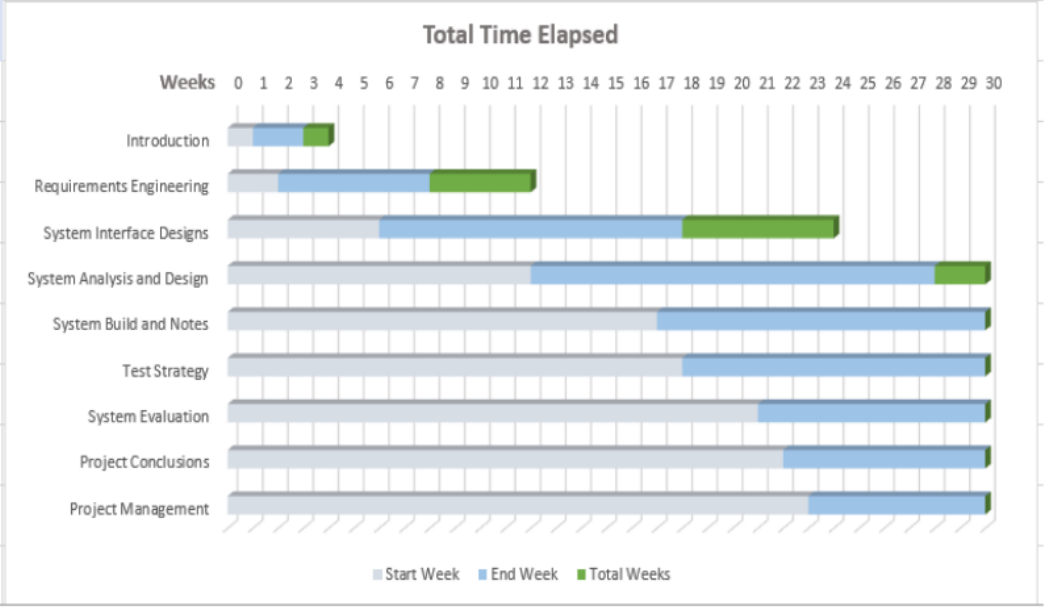


Figure 1. 49 Gantt chart to showcase the time elapsed in the documentation of the project

Code and Implementation

Topics	Start Week	End Week	Total Weeks
User Interface	13	16	3
Administrator	18	22	4
Module Leader	22	24	2
Student	24	28	4

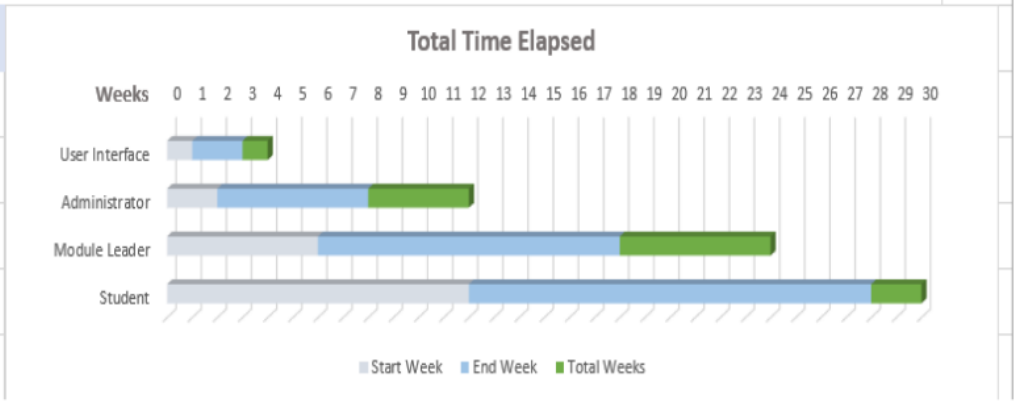


Figure 1. 50. . Gantt chart to showcase the time elapsed in the code and functionalities implementation of the project

9.2. Project Meeting Minutes

Meeting 1:

Date: 22nd November, 2019

Time: 10:30 a.m.

Venue: Room 106, Nami College

In Attendance: All members present

Work Completion: None

Discussion topic: Sort interview questions

Conclusions and plans:

- A few interview questions listed.
- Members divided for different interview topics.
- Interview dates discussed and finalized.

Meeting 2:

Date: 3rd December, 2019

Time: 12:30 p.m.

Venue: Room 205, Nami College

In Attendance: All members present

Work Completion: All interviews completed.

Discussion topic: Division of work

Task performed:

- Work division plans proposed and finalized

Conclusions and plans:

- Start respective work.
- Start with introduction and wireframes

Meeting 3:

Date: 16th December, 2019

Time: 10:30 a.m.

Venue: Meeting Room , Nami College

In Attendance: All members, Mr. Himalaya Kakshapati

Work Completion: Interview questions and introduction documentation

Discussion topic: Work progress update, suggestions and changes

Task performed:

- Discussion about the project plan
- Suggestions on starting the Requirement specifications first by Himalaya sir.

Conclusions and plans:

- To start with requirement specifications and wireframes.

Meeting 4 :

Date: 3rd January, 2020

Time: 02:30 p.m.

Venue: Room 101 , Nami College

In Attendance: All members

Work Completion: Wireframes and up to existing business operations of requirements specifications

Discussion topic: Work progress update, suggestions and changes

Task performed:

- Discussion on the completed tasks and confusions related to functionalities

Conclusions and plans:

- To complete requirement specifications and start with functional requirements.

Meeting 5 :

Date: 27th January, 2020

Time: 10:30 a.m.

Venue: Room 106 , Nami College

In Attendance: All members

Work Completion: Wireframes and up to functional requirements

Discussion topic: Wireframes update

Task performed:

- Discussion on the completed tasks and confusions related to functionalities
- Small change in wireframes

Conclusions and plans:

- To complete the wireframes and start with mockups.

Meeting 6:

Date: 4th February, 2020

Time: 10:30 a.m.

Venue: Meeting Room , Nami College

In Attendance: All members and Mr. Himalaya Kakshapati

Work Completion: Wireframes , few mockups, requirements specifications and System Analysis and Design

Discussion topic: Work progress update, suggestions and changes

Task performed:

- Discussion on the completed tasks and confusions related to functionalities
- Suggestion on changing the wireframes and completing requirements engineering.

Conclusions and plans:

- To complete requirement engineering and wireframes soon.

Meeting 7 :

Date: 20th February, 2020

Time: 11:30 a.m.

Venue: Meeting Room, Nami College

In Attendance: All members and Mr. Himalaya Kaskshapati

Work Completion: Wireframes, mockups, requirements engineering and System Analysis and Design

Discussion topic: Work progress update, suggestions and changes

Task performed:

- Suggestion on starting the UI design of the system/ Code

Conclusions and plans:

- To complete the pending work and start with the code and implementation.

Meeting 8 :

Date: 27th February, 2020

Time: 11:30 a.m.

Venue: Meeting Room, Nami College

In Attendance: All members and Mr. Himalaya Kaskshapati

Work Completion: Wireframes, mockups, requirements engineering and System Analysis and Design, UI design

Discussion topic: Work progress update, suggestions and changes

Task performed:

- Suggestion on implementing better ways of coding and changing the UI

Conclusions and plans:

- To change the UI of the system and complete the admission process

Meeting 9:

Date: 5th March, 2020

Time: 11:30 a.m.

Venue: Meeting Room, Nami College

In Attendance: All members and Mr. Himalaya Kaskshapati

Work Completion: Wireframes, mockups, requirements engineering and System Analysis and Design, UI design, admission process

Discussion topic: Work progress update, suggestions and changes

Task performed:

- Suggestion on speeding up the work and implement more functionalities

Conclusions and plans:

- To complete the admin area functionalities

Meeting 10:

Date: 8th April, 2020

Time: 3:00 p.m.

Venue: via Email

In Attendance: None

Work Completion: Admin area, Student area and documentation up to topic 4

Discussion topic: Work progress update, suggestions and changes

Task performed:

- Present report to sir for amendments, changes and suggestions.

Conclusions and plans:

- To make a few changes in the report like classes and objects, white spaces and interview answers elaboration.

Meeting 11:

Date: 20th April, 2020

Time: 3:00 p.m.

Venue: via Skype

In Attendance: All members

Work Completion: Admin area, Student area and documentation up to topic 4

Discussion topic: Work progress update, suggestions and changes

Task performed:

- Discussion on the remaining topics and due to the current situation the delay in project completion.

Conclusions and plans:

- Complete the module leader area and start with testing.

Meeting 12:

Date: 2nd May, 2020

Time: 3:00 p.m.

Venue: via Skype

In Attendance: All members and Mr. Himalaya Kakshapati

Work Completion: Admin area, Student area, Module Leader area and documentation up to topic 4+ Project Management

Discussion topic: Work progress update, suggestions and changes

Task performed:

- Present the system to sir for suggestions and amendments.
- Decision on choosing the 2nd submission deadline for submission due to incomplete tasks.

Conclusions and plans:

- Suggestion to make drastic changes in the UI and add more functionalities.

Meeting 13:

Date: 30th May, 2020

Time: 12:00 p.m.

Venue: via Skype

In Attendance: All members

Work Completion: Admin area, Student area, Module Leader and documentation up to topic 4, change in UI

Discussion topic: Work progress update, suggestions and changes

Task performed:

- Discussion on adding a few functionalities and finalizing the UI.

Conclusions and plans:

- To complete the remaining functionalities and start with testing.

Meeting 14:

Date: 15th June, 2020

Time: 2:00 p.m.

Venue: via Skype

In Attendance: All members

Work Completion: Admin area, Student area, Module Leader and documentation up to topic 4

Discussion topic: Work progress update, suggestions and changes

Task performed:

- Discuss on changing a few functionalities and start with remaining documentation.

Conclusions and plans:

- To complete the project.

Meeting 15:

Date: 27th June, 2020

Time: 4:00 p.m.

Venue: via Skype

In Attendance: All members

Work Completion: Completed documentation and system.

Discussion topic: Work progress update, suggestions and changes

Task performed:

- Discuss on all the work done and if needed add changes.

Conclusions and plans:

- Make a few changes in the system.

9.3. Project Quality Plan/ Strategy

The project quality plan/ strategy (PQP) consists of many components that contribute in analysing the quality plan of a project. Quality is the expectation of stakeholders. The more the demands have been met, the more qualitative the project is considered. The initial plan of any project developer is to complete as much as the expectations of the client but incorporating all the functionalities could be unachievable due to either time constraint or inflexibility.

The functionalities completed in the system and the strategies and importance level for the functionalities are as follows:

S. No.	Functionality	Importance level	Completion status
Administrator			
1.	Admission process using excel spreadsheets and automatic announcement via email	**Very High**	Completed
7.	Adding administrators, module leaders to the system	**Very High**	Completed
3.	Assigning PAT to the students	**Moderate**	Completed
4.	Assigning course to the students	**High**	Completed

5.	Changing the status of the students	**Very High**	Completed
6.	Assigning a module to a module leader	**High**	Completed
7.	Adding new courses, modules	**High**	Completed
8.	Add announcements	**High**	Completed
9.	Edit and delete any added students/module leaders/courses/modules	**Very High**	Completed
10.	Add personal notes	**Low**	Completed
Module Leader			
11.	Add module contents	**Very High**	Completed
12.	Add assignments	**Very High**	Completed
13.	Add grades	**Very High**	Completed
14.	Send email to students for PAT	**Low**	Completed
15.	Send warning email for low attendance	**High**	Completed
16.	Add attendance	**High**	Completed

17.	Add announcements	**Moderate**	Completed
Student			
18.	Submit assignment	**High**	Completed
19.	View grades	**High**	Completed
20.	Send email to PAT	**Low**	Completed

9.4. Project Work Log

Task	Individual Contribution						
Introduction	Aditya Jha	Anish Gelal	Grishma Duwadi	Manish Thapa	Prekshya Neupane	Sagun Rupakheti	Sushank Aryal
Project Background			100%				
Project Aims and Objectives			100%				
Project Development Methodology			100%				
Requirements Engineering	Aditya Jha	Anish Gelal	Grishma Duwadi	Manish Thapa	Prekshya Neupane	Sagun Rupakheti	Sushank Aryal
Design Interview Questions	10%					90%	

Interviews	14%	14%	14%	14%	14%	14%	14%
Write Up Interview Answer				100%			
Comparable Software System Review				80%	20%		
Development Relevant Legislation				80%	20%		
Requirements Specification	Aditya Jha	Anish Gelal	Grishma Duwadi	Manish Thapa	Prekshya Neupane	Sagun Rupakheti	Sushank Aryal
Problem Domain Description		20%				80%	
Existing Bussiness Operation		20%				80%	

Summary of Existing business limitations		20%				80%	
Functional Requirements						100%	
Performance Requirements						100%	
Design Constraints						100%	
Commercial Constraints						100%	
System Analysis and Design	Aditya Jha	Anish Gelal	Grishma Duwadi	Manish Thapa	Prekshya Neupane	Sagun Rupakheti	Sushank Aryal
Preliminary Design Stages							

Textual Analysis							100%
Significant Event Analysis						100%	
Commands Queries and Constraints			50%			50%	
Detailed static system designs						100%	
Detailed dynamic system designs						100%	
System database design		20%		80%			
System Interface Designs	Aditya Jha	Anish Gelal	Grishma Duwadi	Manish Thapa	Prekshya Neupane	Sagun Rupakheti	Sushank Aryal

Wireframes	100%						
Navigation diagrams	70%					30%	
System screen mock-ups	100%						
System Activity event diagram						100%	
System Build and technical notes	Aditya Jha	Anish Gelal	Grishma Duwadi	Manish Thapa	Prekshya Neupane	Sagun Rupakheti	Sushank Aryal
On-going Testing Methodology						100%	
Final System Interface Displays						100%	

System Build and technical notes	Aditya Jha	Anish Gelal	Grishma Duwadi	Manish Thapa	Prekshya Neupane	Sagun Rupakheti	Sushank Aryal
Technically Difficult Code Sections				50%		50%	
On-going Testing Methodology							100%
Final System Interface Displays		50%	50%				
Test Strategy	Aditya Jha	Anish Gelal	Grishma Duwadi	Manish Thapa	Prekshya Neupane	Sagun Rupakheti	Sushank Aryal
Overview of Test Strategy					50%	50%	
Sample Test Results						100%	

System Evaluation	Aditya Jha	Anish Gelal	Grishma Duwadi	Manish Thapa	Prekshya Neupane	Sagun Rupakheti	Sushank Aryal
Usability Evaluation Strategy							100%
System Pilot Trials						100%	
System Trial Results			100%				
Project Conclusions	Aditya Jha	Anish Gelal	Grishma Duwadi	Manish Thapa	Prekshya Neupane	Sagun Rupakheti	Sushank Aryal
							100%
Project Management	Aditya Jha	Anish Gelal	Grishma Duwadi	Manish Thapa	Prekshya Neupane	Sagun Rupakheti	Sushank Aryal

Project Gantt Chart/WBDS/Activity List						100%	
Project Meeting Minutes						100%	
Project Quality Plan/ Strategy						100%	
Code and implementation	Aditya Jha	Anish Gelal	Grishma Duwadi	Manish Thapa	Prekshya Neupane	Sagun Rupakheti	Sushank Aryal
System Development and UI	20%	10%		55%		15%	

10. References

1. Subjectmoney(2020) Gantt Chart Excel Tutorial - How to make a Basic Gantt Chart in Microsoft Excel 2013.Youtube[online].Available from https://www.youtube.com/watch?v=-oD50HSBBBI&t=177s

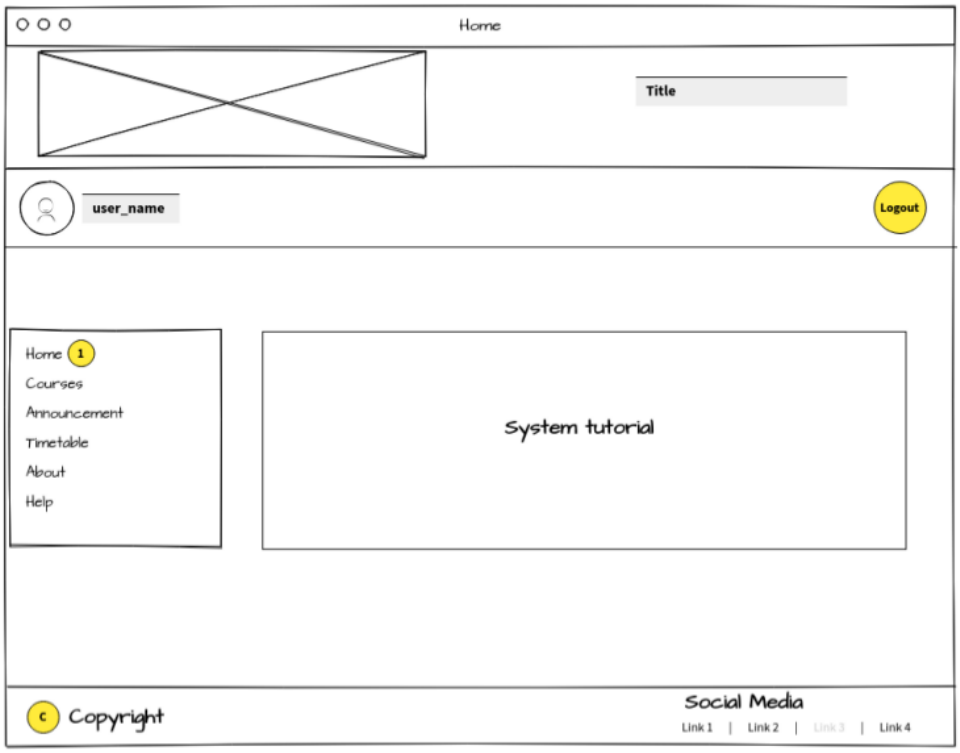
2. The Editorial Team(2020) Defining the 10 Project Quality Plan Components of Project Management. nutcache[online]. Available from:https://www.nutcache.com/blog/defining-10-project-quality-plan-components-project-management/

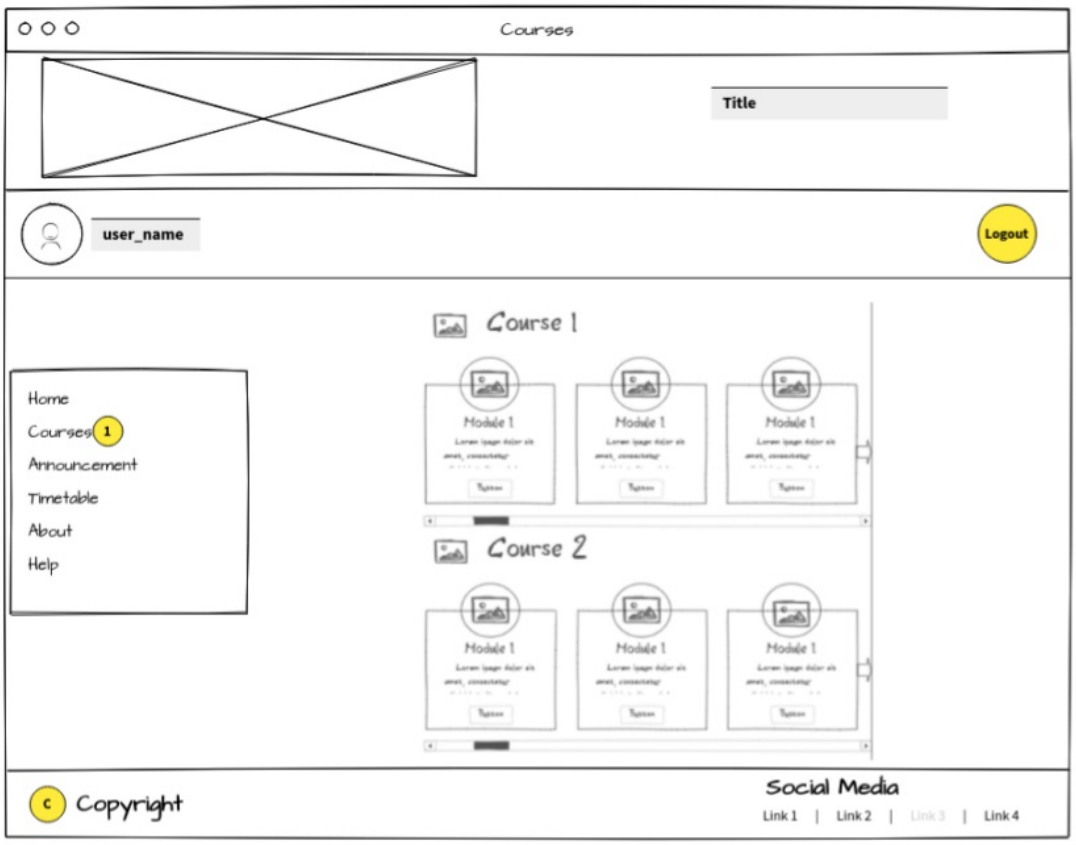
3. GOV.UK(2018).Data Protection.GOV.UK[online]. Available from: https://www.gov.uk/data-protection

4. legislation.gov.uk(1990).Computer Misuse Act 1990. legislation.gov.uk [online]. Available from: http://www.legislation.gov.uk/ukpga/1990/18/contents

5. GOV.UK(2016).Digital Economy bill 2016.GOV.UK[online]. Available from: https://www.gov.uk/government/collections/digital-economy-bill-2016

11. Appendix


Home
Title
user_name
Logout
Home
1
Courses
Announcement
Timetable
About
Help
System tutorial
c
Copyright
Social Media
Link 1 | Link 2 | Link 3 | Link 4


Courses
Title
user_name
Logout
Home
Courses 1
Announcement
Timetable
About
Help
Course 1
Module 1
Module 1
Module 1
Course 2
Module 1
Module 1
Module 1
Copyright
Social Media
Link 1 | Link 2 | Link 3 | Link 4

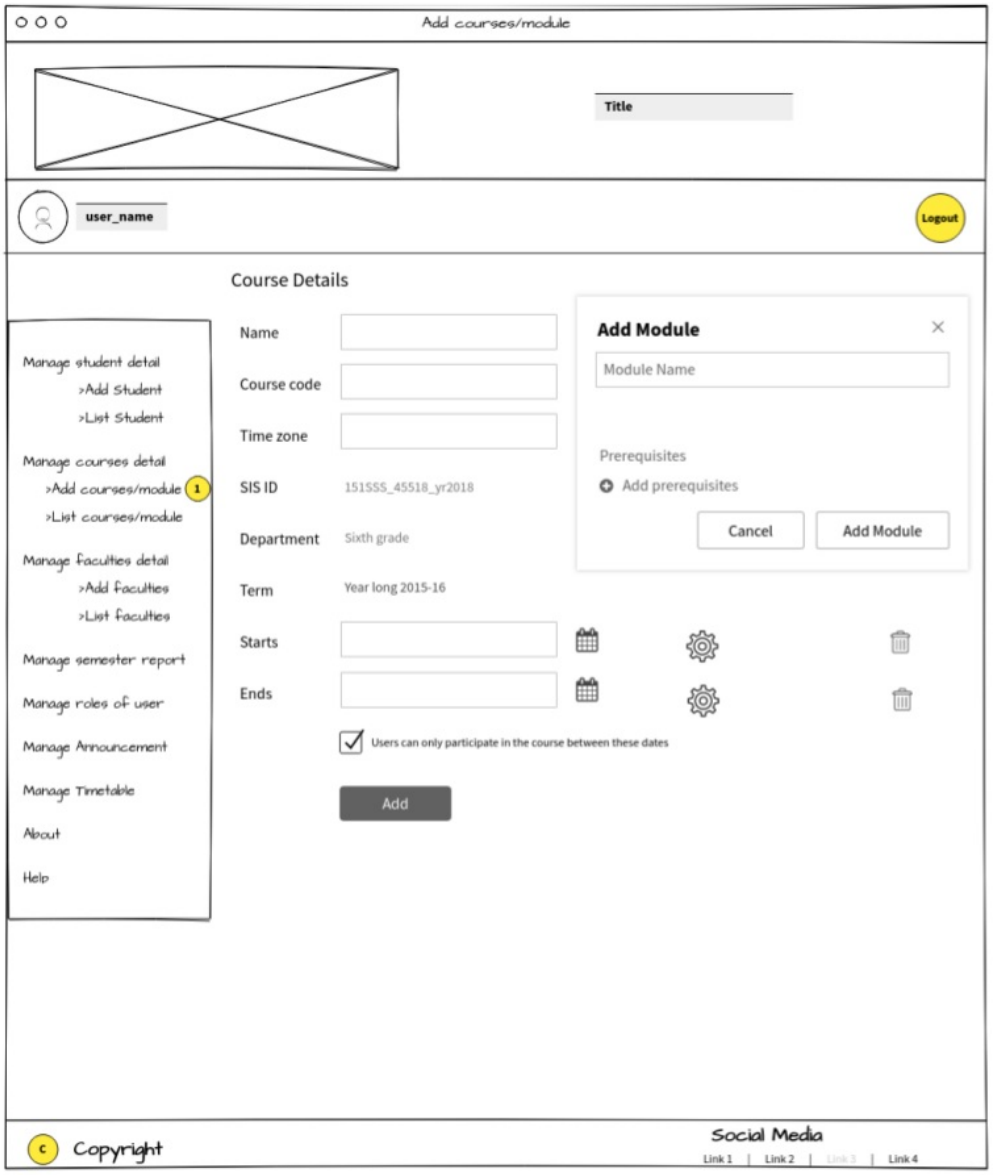


Figure 1. 51. Wireframe showing edit course page

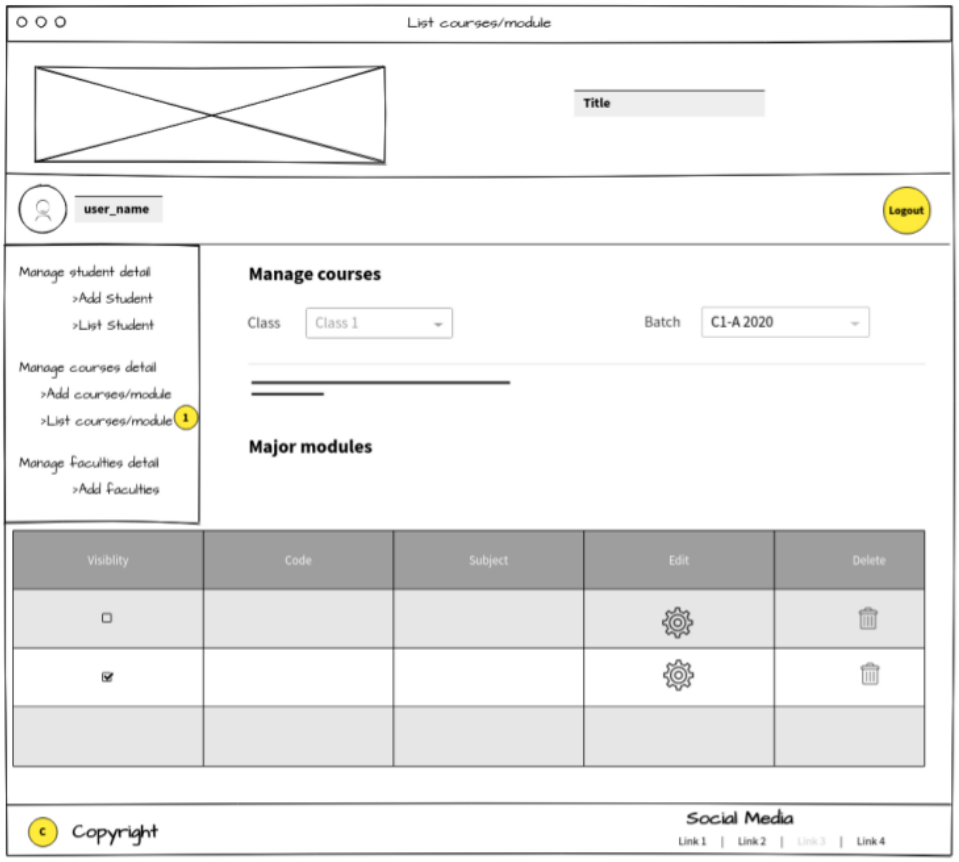


Figure 1. 52. Wireframe showing list of module/courses

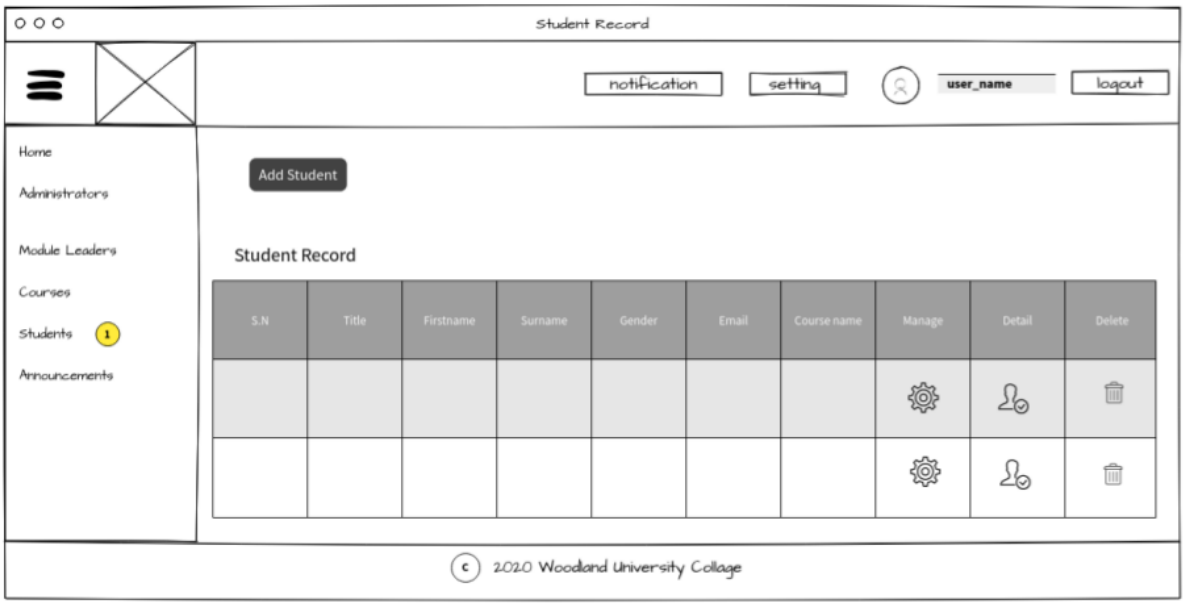


Figure 1. 53. Wireframe showing students records to edit

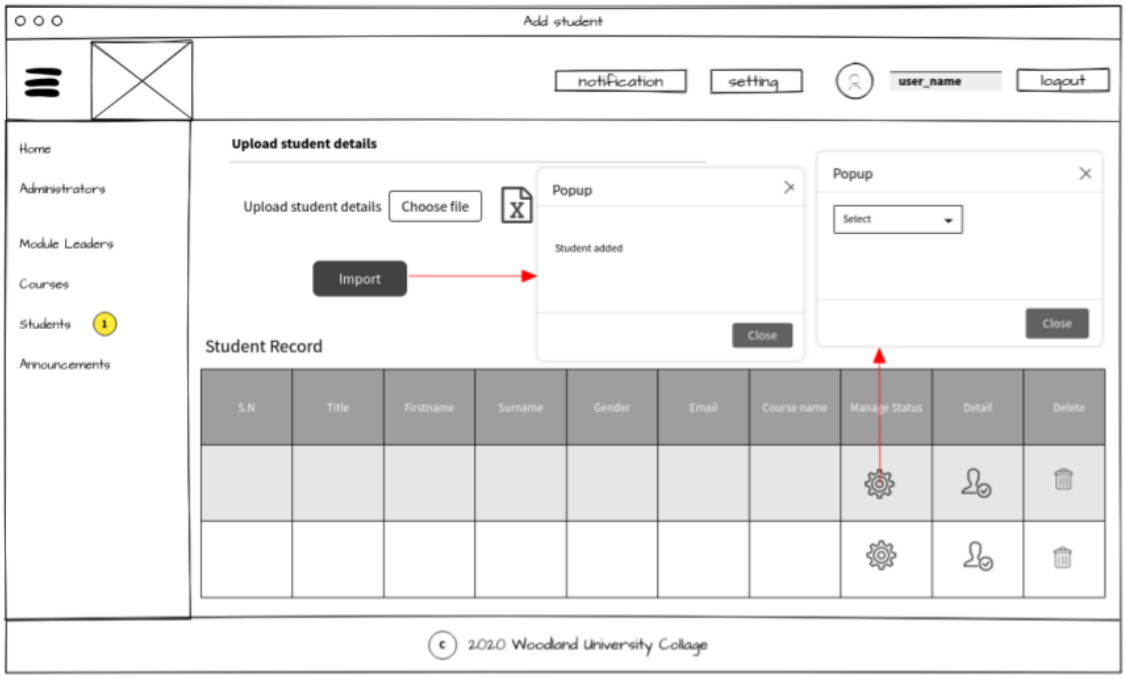
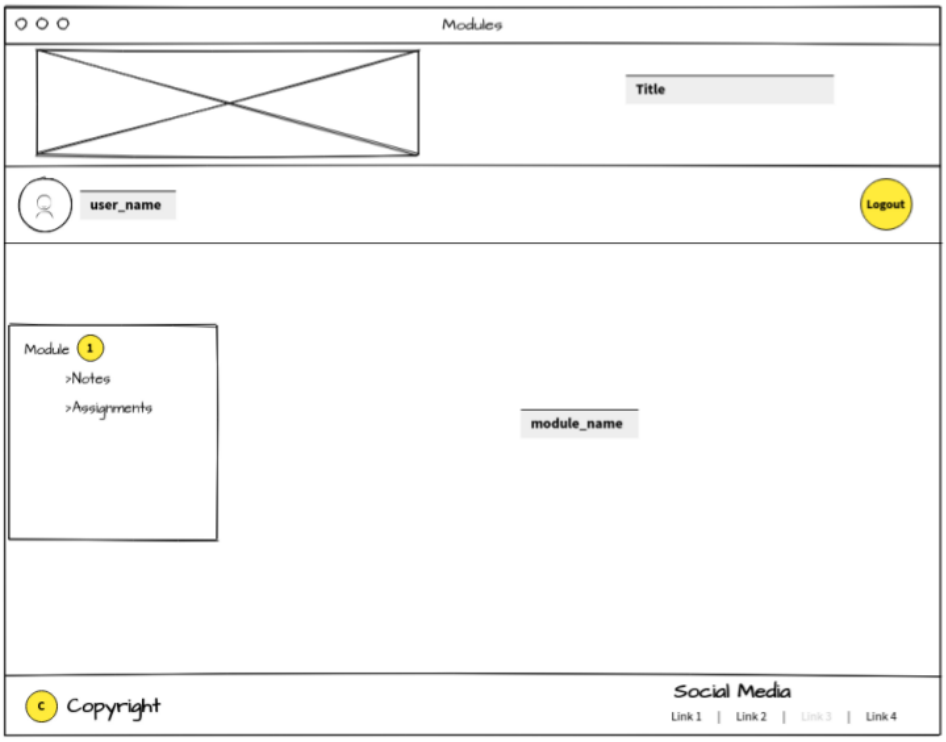
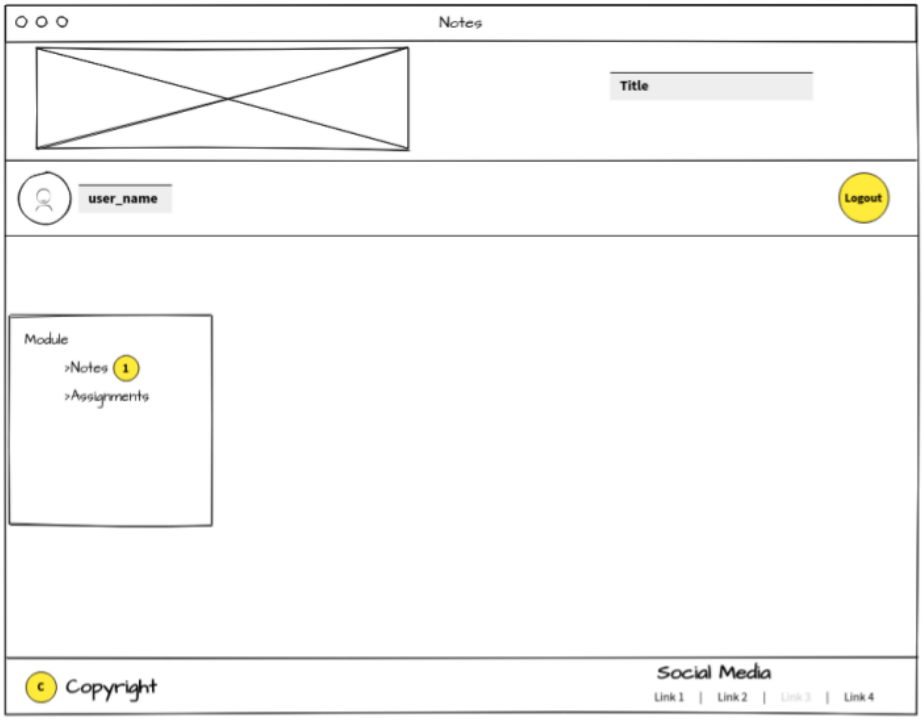


Figure 1. 54. Wireframe showing admission process


Modules
Title
user_name
Logout
Module 1
>Notes
>Assignments
module_name
Copyright
Social Media
Link 1 | Link 2 | Link 3 | Link 4


Notes
Title
user_name
Logout
Module
>Notes
1
>Assignments
c
Copyright
Social Media
Link 1 | Link 2 | Link 3 | Link 4

Student Detail
notification
setting
user_name
logout
Home
Administrators
Module Leaders
Courses
Students
1
Announcements
XYZ
Class
Class 1
Batch
A 2020
Admn no
S1
Roll number
C101
Admission date
24 Aug 2019
Date of birth
23 sep
Blood group
O+
Gender
Male
Nationality
United states
Mother tongue
English
Mobile
-
E-mail
-
SMS alerts
-
Biometric ID
-
Name
-
Username
-
Edit
PDF report
© 2020 Woodland University Collage

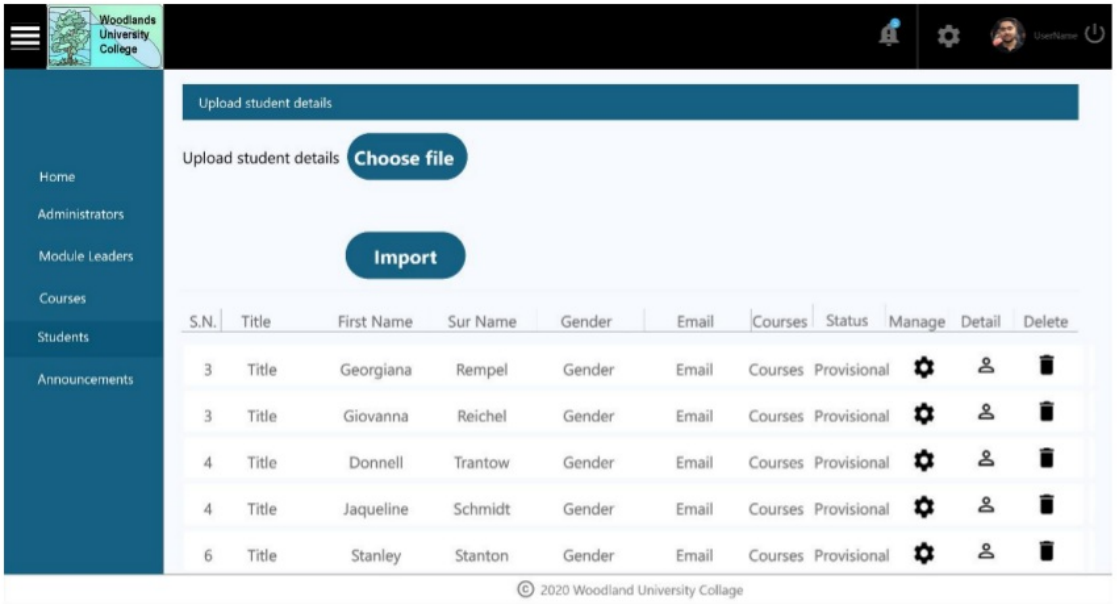


Figure 1. 55. The admin chooses to upload a file

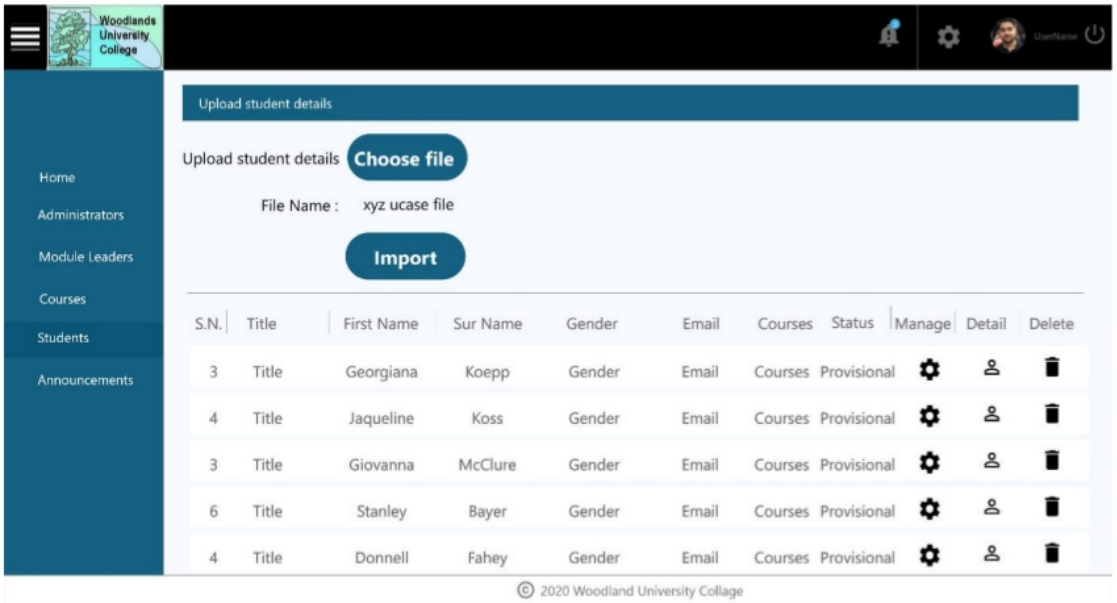


Figure 1. 56 The admin selects a UCAS file to upload

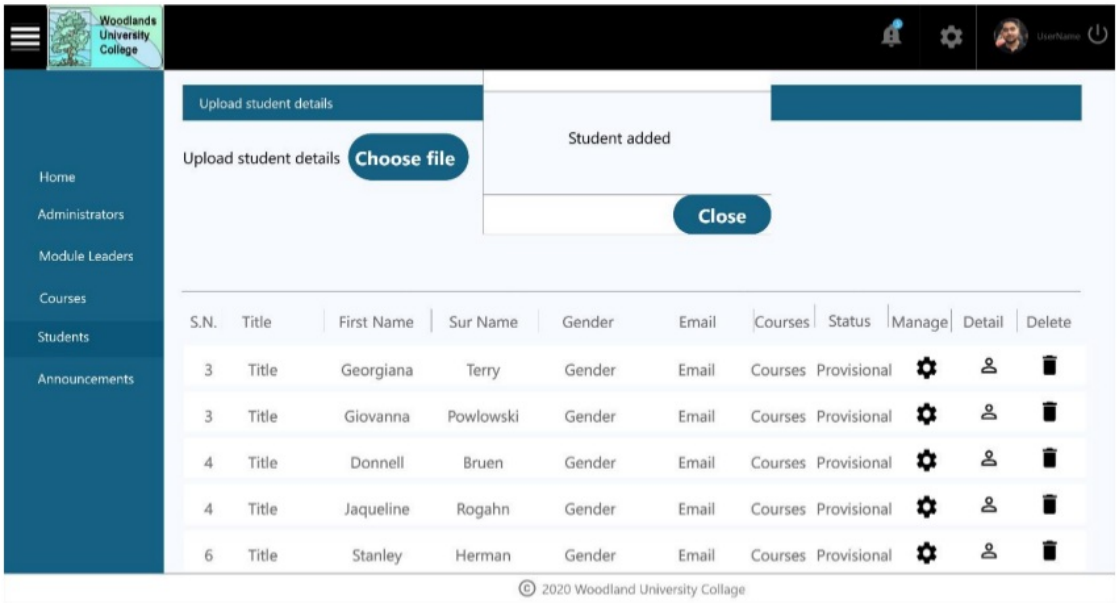


Figure 1. 57 The UCAS file is uploaded

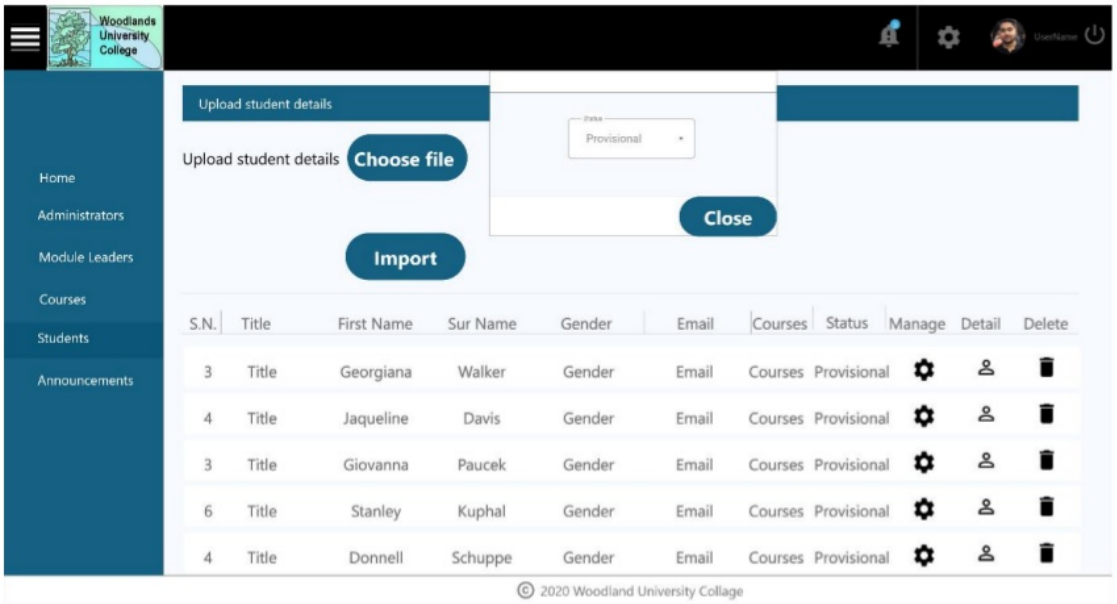


Figure 1. 58. The UCAS file is uploaded

Figure 1. 59. Student dashboard

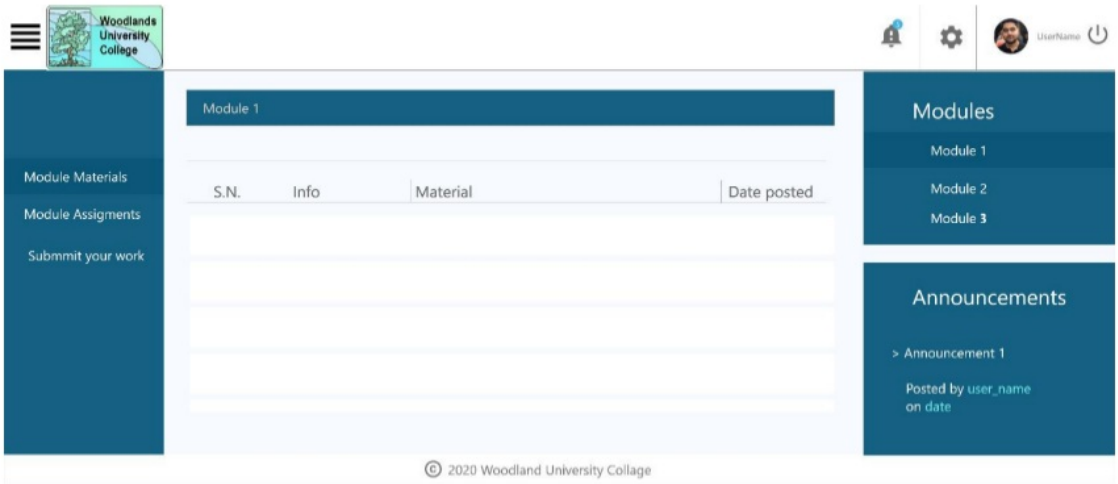


Figure 1. 60. Student views module materials

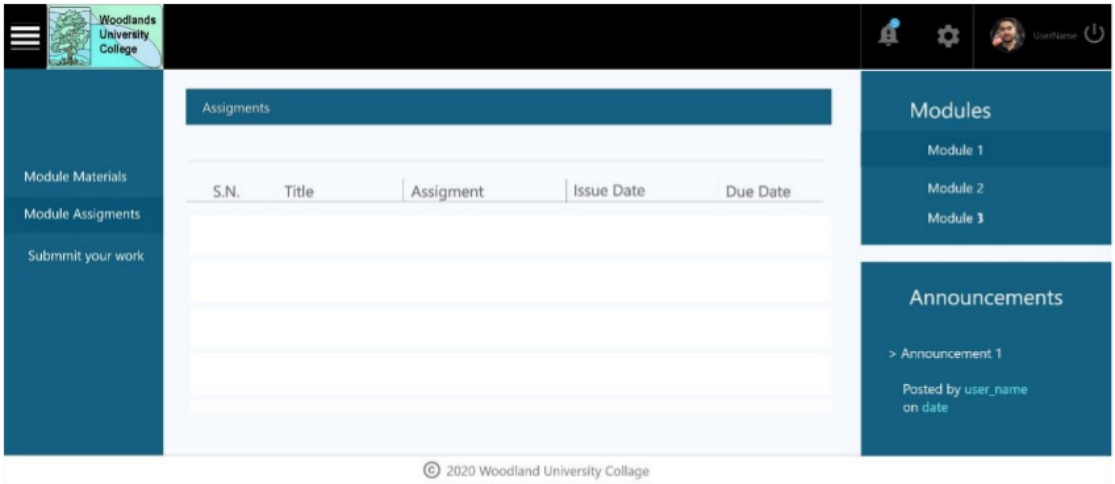


Figure 1. 61. Student views module assignments

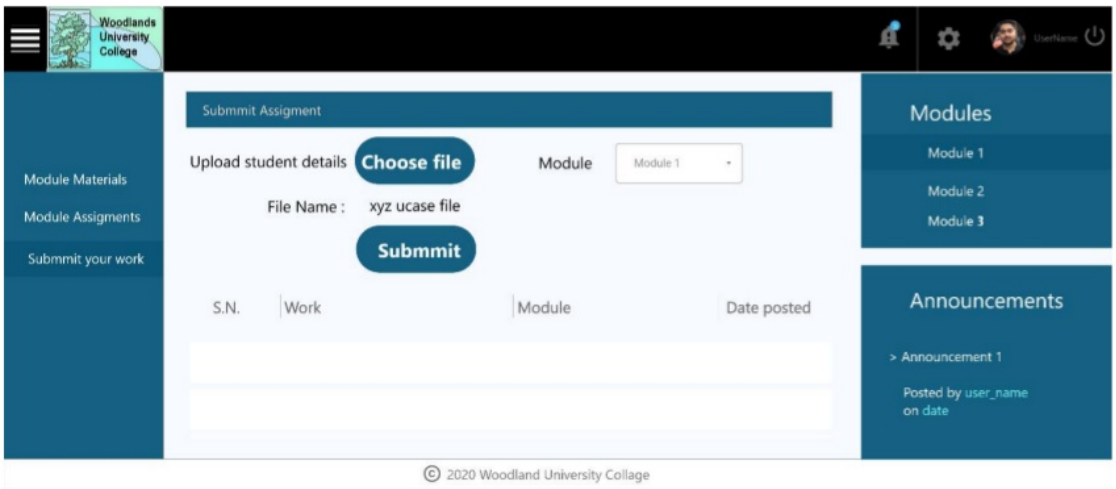


Figure 1. 62. Student uploads assignment

CSY2027 GROUP PROJECT

Group E

Group Members

- Aditya Jha
- Anish Gelal
- Grishma Duwadi
- Manish Thapa
- Prekshya Neupane
- Sagun Rupakheti
- Sushank Aryal

Project Background

- Client- Woodlands University College (WUC)
- To digitalize the course management system of Woodlands University College
- A well managed database system
- Designated functionalities for all 3 user groups
- User groups: administrator, module leader, student

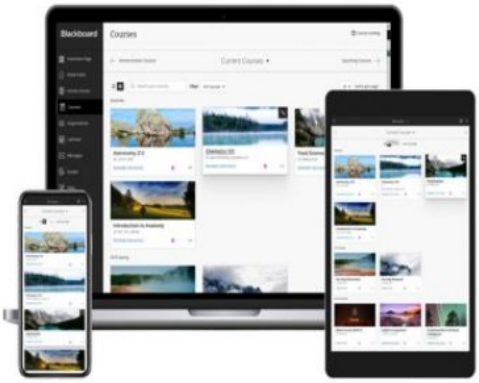
PROJECT FLOW

- Interview with different user groups
- Listing and documentation of all the functionalities
- A draft of the system as wireframes and mockups
- Implement the required functionalities
- Test the system
- Project Management

Comparable systems

Moodle

Positive aspects	Negative Aspects
Device compatible	Not user-friendly
Customizable UI	Complex and technical UI
Vast and informative	High maintenance cost
Notification system	Feels heavy to use
Suitable color combinations	



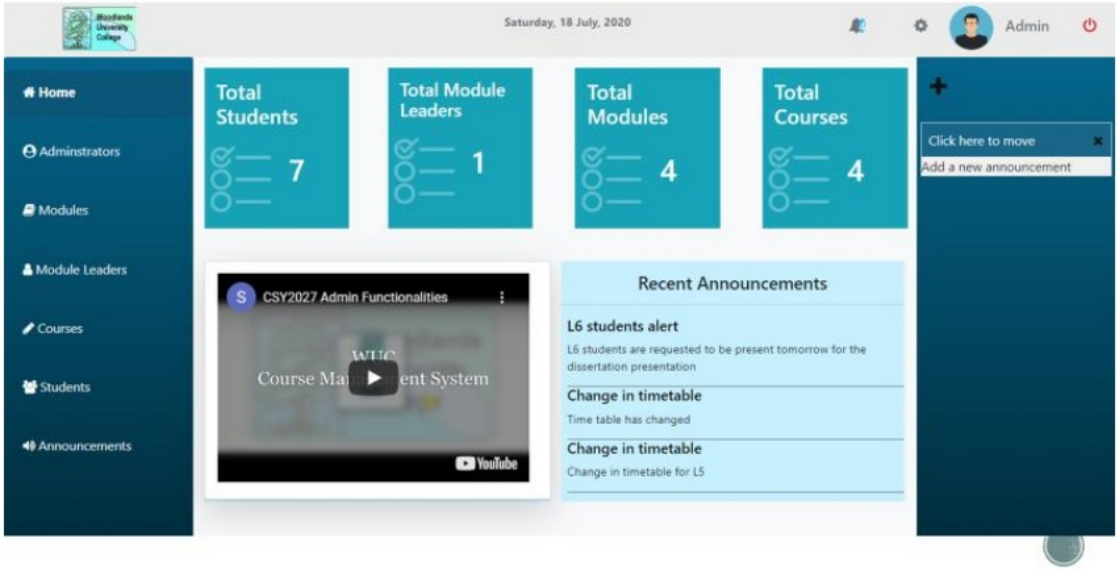
BlackBoard

Positive aspects	Negative Aspects
User friendly	No notification system
Easy Navigation and UI	Not suitable color combinations
Low maintenance cost	Poor device compatibility
Clean look	
Feels light and lag free to use	

USER GROUPS AND FUNCTIONALITIES

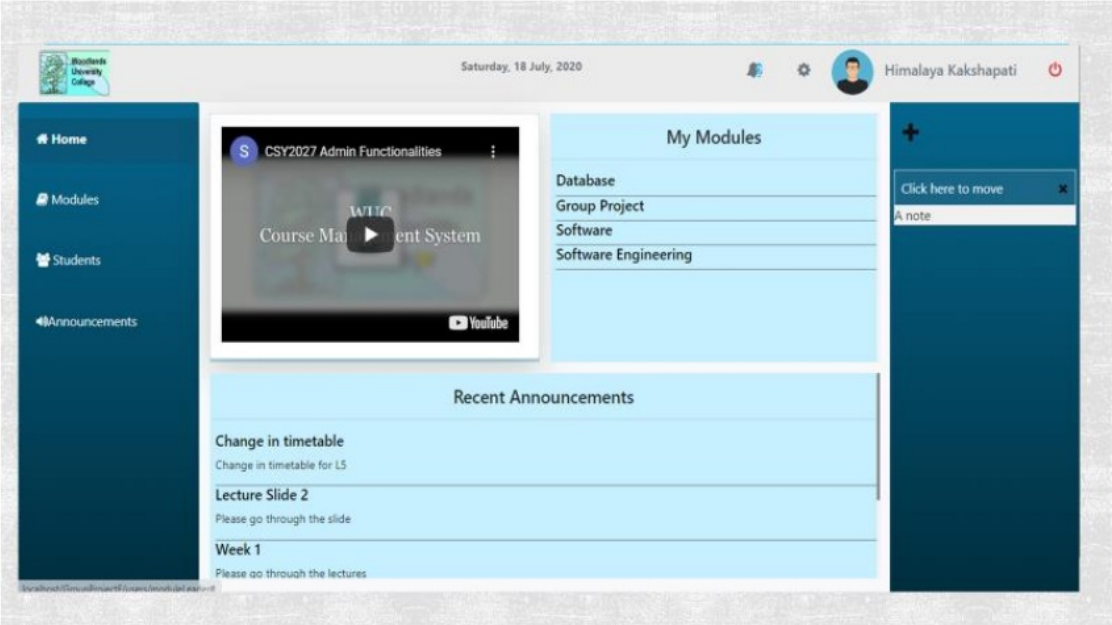
- Admin management
- Module leader management
- Admission management
- Course Management
- Modules Management
- Time-table management
- Diary management
- Announcement Management

Administrator



- Module-content management
- Assignment management
- Grades Management
- Attendance Management
- Announcement Management
- PAT communication

Module Leader



View module content

Submit assignments

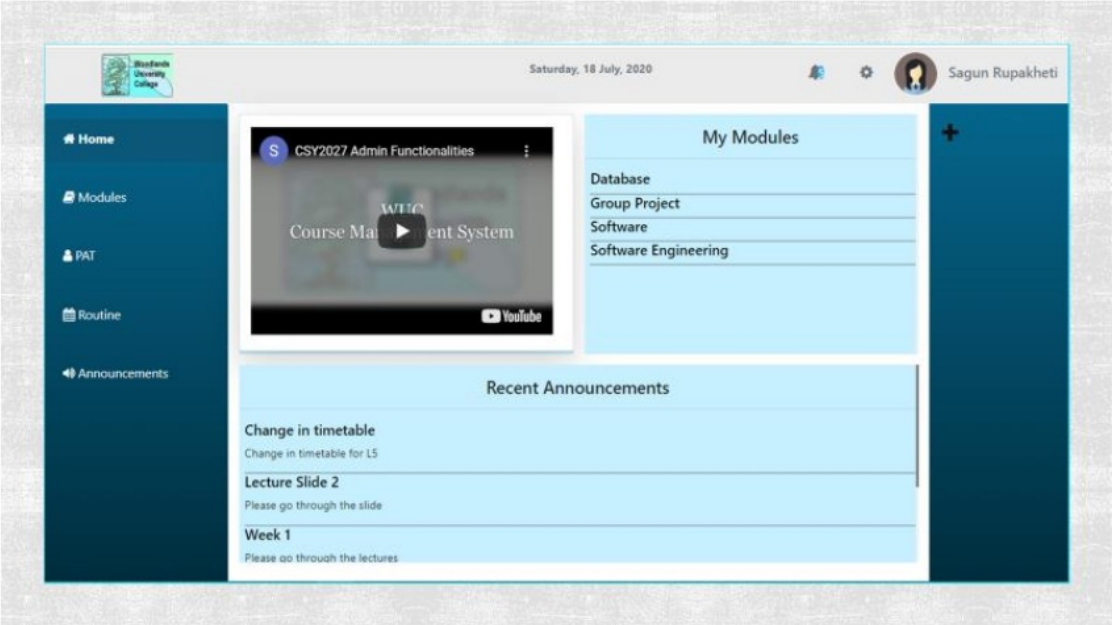
View grades

View time-table

Diary Management

PAT communication

Student



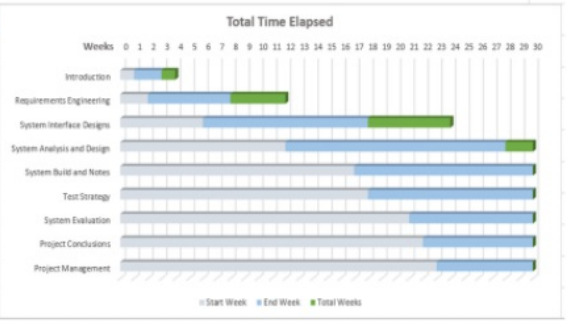
Code and Implementation

Topics	Start Week	End Week	Total Weeks
User Interface	13	16	3
Administrator	18	22	4
Module Leader	22	24	2
Student	24	28	4

Gantt Chart for implementation

Documentation

Topics	Start Week	End Week	Total Weeks
Introduction	1	2	1
Requirements Engineering	2	6	4
System Interface Designs	6	12	6
System Analysis and Design	12	16	4
System Build and Notes	17	18	1
Test Strategy	18	21	3
System Evaluation	21	22	1
Project Conclusions	22	23	1
Project Management	23	26	3



Gantt Chart for documentation

ANY QUESTIONS?

THANK-YOU

Group Project group E technical report

GRADEMARK REPORT

FINAL GRADE

/100

GENERAL COMMENTS

Instructor

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAGE 19

PAGE 20

PAGE 21

PAGE 22

PAGE 23

PAGE 24

PAGE 25

PAGE 26

PAGE 27

PAGE 28

PAGE 29

PAGE 30

PAGE 31

PAGE 32

PAGE 33

PAGE 34

PAGE 35

PAGE 36

PAGE 37

PAGE 38

PAGE 39

PAGE 40

PAGE 41

PAGE 42

PAGE 43

PAGE 44

PAGE 45

PAGE 46

PAGE 47

PAGE 48

PAGE 49

PAGE 50

PAGE 51

PAGE 52

PAGE 53

PAGE 54

PAGE 55

PAGE 56

PAGE 57

PAGE 58

PAGE 59

PAGE 60

PAGE 61

PAGE 62

PAGE 63

PAGE 64

PAGE 65

PAGE 66

PAGE 67

PAGE 68

PAGE 69

PAGE 70

PAGE 71

PAGE 72

PAGE 73

PAGE 74

PAGE 75

PAGE 76

PAGE 77

PAGE 78

PAGE 79

PAGE 80

PAGE 81

PAGE 82

PAGE 83

PAGE 84

PAGE 85

PAGE 86

PAGE 87

PAGE 88

PAGE 89

PAGE 90

PAGE 91

PAGE 92

PAGE 93

PAGE 94

PAGE 95

PAGE 96

PAGE 97

PAGE 98

PAGE 99

PAGE 100

PAGE 101

PAGE 102

PAGE 103

PAGE 104

PAGE 105

PAGE 106

PAGE 107

PAGE 108

PAGE 109

PAGE 110

PAGE 111

PAGE 112

PAGE 113

PAGE 114

PAGE 115

PAGE 116

PAGE 117

PAGE 118

PAGE 119

PAGE 120

PAGE 121

PAGE 122

PAGE 123

PAGE 124

PAGE 125

PAGE 126

PAGE 127

PAGE 128

PAGE 129

PAGE 130

PAGE 131

PAGE 132

PAGE 133

PAGE 134

PAGE 135

PAGE 136

PAGE 137

PAGE 138

PAGE 139

PAGE 140

PAGE 141

PAGE 142

PAGE 143

PAGE 144

PAGE 145

PAGE 146

PAGE 147

PAGE 148

PAGE 149

PAGE 150

PAGE 151

PAGE 152

PAGE 153

PAGE 154

PAGE 155

PAGE 156

PAGE 157

PAGE 158

PAGE 159

PAGE 160

PAGE 161

PAGE 162

PAGE 163

PAGE 164

PAGE 165

PAGE 166

PAGE 167

PAGE 168

PAGE 169

PAGE 170

PAGE 171

PAGE 172

PAGE 173

PAGE 174

PAGE 175

PAGE 176

PAGE 177

PAGE 178

PAGE 179

PAGE 180

PAGE 181

PAGE 182

PAGE 183

PAGE 184

PAGE 185

PAGE 186

PAGE 187

PAGE 188

PAGE 189

PAGE 190

PAGE 191

PAGE 192

PAGE 193

PAGE 194

PAGE 195

PAGE 196

PAGE 197

PAGE 198

PAGE 199

PAGE 200

PAGE 201

PAGE 202

PAGE 203

PAGE 204

PAGE 205

PAGE 206

PAGE 207

PAGE 208

PAGE 209

PAGE 210

PAGE 211

PAGE 212

PAGE 213

PAGE 214

PAGE 215

PAGE 216

PAGE 217

PAGE 218

PAGE 219

PAGE 220

PAGE 221

PAGE 222

PAGE 223

PAGE 224

PAGE 225

PAGE 226

PAGE 227

PAGE 228

PAGE 229

PAGE 230

PAGE 231