**The Design and Development**

**Of**

**Course Management Software System**

***By: Everest***

**Submission date:**

**Submission id:**

**File name:**

**Word count:**

**Character count:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **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**  Inshan KC - 20416224  Pankaj Badu – 20416237  Milan Khatiwada-20416231  Ronak Bastola- 20416250 | | | | | | **Tutor: Suresh Gautam** | |
| 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 Content

1 [Introduction](#Intrdouction)

* 1. [Project Background](#ProjectBackground)
  2. [Project Aims and Objectives](#ProjectAimsandObjectives)
  3. [Project Development Methodology](#ProjectDevelopmentMethodology)

1. [Requirement Engineering](#RequirementEngineering)
   1. [Elicitation Activities](#ElicitationActivities)
      1. [Interview Plans](#InterviewPlans)
      2. [Interview Findings](#InterviewFindings)
      3. [Other problem domain research](#Otherproblemdomainresearch)
         1. [Comparable software system review](#Comparablesoftwaresystemreview)
         2. [NILE](#NILE)
         3. [Coursera](#Coursera)

2.1.3.2 [Development Relevant Legislation](#DevelopmentRelevantLegislation)

2.1.3.2.1 [Data Protection Right](#DataProtectionRight)

2.1.3.2.2 [Digital Economy Act](#DigitalEconomyAct)

2.1.3.2.3 [Computer Misuse Act](#ComputerMisuseAct)

2.1.3.3 [Any other relevant problem domain investigation data](#Anyotherrelevantproblemdomaininvestment)

* 1. [Requirement Specification](#RequirementsSpecification)
     1. [Problem Domain Description](#ProblemDomainDescription)
        1. [Existing business operation](#ExistingbusinessOperation)
           1. [UCAS and letter](#UCASandletter)
           2. [ID generation and student folder](#IDgenerationandstudentfolder)
        2. [Summary of existing business limitations](#Summaryofexistingbusinesslimitations)
     2. [Functional requirements](#FunctionalRequirements)
        1. [Required features of the system](#RequiredFeaturesofthesystem)
           1. [Database and backup](#Databaseandbackup)
           2. [Students record](#Studentsrecord)
           3. [Staffs record](#Staffrecords)
           4. [Course records](#Courserecords)
           5. [Module management](#Modulemanagement)
           6. [Assignment records](#Assignmentmanagement)
           7. [Attendance management](#Attendancemanagement)
           8. [Personal tutor management](#PersonalTutormanagement)
           9. [Grades management](#Gradesmanagement)
           10. [Module content management](#Modulecontentmanagement)
           11. [Announcement management](#Announcementmanagement)
        2. [Access rights/ level of access](#Accessrightslevelofaccess)
           1. [Administrator, staff and student management](#Administratorsrecordmanagement)
           2. [Course management](#Coursemanagement)
           3. [Module and content management](#ModulecontentManagement)
           4. [Assignment management](#Assignmentmanagementt)
           5. [Grade management](#Grademanagement)
           6. [Attendance management](#Attendancemanagementt)
           7. [Personal tutor management](#Personaltutormanagementt)
        3. [Output formats for onscreen displays](#LowAttendanceGradespblishedAssign)
           1. Enrolled to the system
           2. Low attendance
           3. Grades published
           4. Assignment published
           5. Assignment submitted
           6. Pat meeting schedule
     3. [Performance requirements](#Performance)
        1. [Speed](#Speed)
        2. [Capacity](#Capacity)
        3. [Reliability](#Reliability)
     4. [Design constraints](#Designconstraints)
     5. [Commercial constraints](#CommercialConstraints)

1. [System analysis and design](#Systemanalysisanddesign)
   1. [Preliminary design stages](#Preliminarydesignstages)
      1. [Textual analysis](#Textualanalysis)
      2. [Significant event analysis](#Systemanalysisanddesign)
      3. [Commands queries and constraints](#Commandsqueriesandconstraints)
   2. [Detailed static system designs](#DetailedStaticSystemDesign)
      1. [First draft BON system architecture diagram](#FirstdraftBONsystemarchitecturediag)
      2. [Use case diagram](#Usecasediagram)
         1. Administrators use case diagram
         2. Module leader use case diagram
         3. Student use case diagram
      3. [Use case documentation](#usecase)
         1. UC1 administrator tab
         2. UC module leader tab
         3. UC3 announcement tab
      4. [Class diagram](#ClassDiagram)
   3. [Detailed dynamic system designs](#Detaileddynamicsystemdesigns)
      1. [System sequence diagram](#Systemsequencediagram)
      2. System collaboration/ communication diagrams
   4. System database design
      1. E-R model
      2. Attribute listings
2. System interface designs
   1. [Draft interface designs](#Draftinterfacedesigns)
      1. [Wireframes](#Wireframes)
      2. [System navigation diagram](#Systemnavigationdiagram)
      3. System screen mock-ups
      4. [System activity event diagrams](#Systemactivityevent)

Abstract

Woodland University has encountered many problems related to the management of their teachers and courses to teach. They need a lot of time to capture huge amounts of data where the user must memorize all the features of the tool and it is not easy and can be the results of various errors which causes delays in their work. Woodland university decided to change its system from clerical to fully digitalized and modernized system. Its main aim is to automate the whole school management system. To know how they want the system to function and what the system should have, we had interviews with Mr. Adam Blake, Dr. Simon White, Mr. Mark Williams, and Dr. Raj Singh.

Based on the outcome of the interview, the functionalities were found out and the project was started along with the designation of UI and use case diagram with the class and system event diagram with sequence diagram. Coursera was taken as reference. We propose a system that shall enable a school management system interfaced with a computer to be managed remotely using personal computers. A client running on the user’s computer would connect to a central server, over an Internet connection. The user could then use a HTTP protocol from a personal computer to connect to the central server, and issue requests for borrowing books. In this report we describe a system – the web enabled software that helps in the management of university.

The project is web based that was developed by using HTML, CSS, JavaScript, and bootstrap at the front end and PHP at the back end based on three tire architecture. The methodology used in this project is Object oriented system analysis and design methodology and software development model methodology. The system is developed based on software development life cycle (SDLC) which includes requirement analysis, system analysis, system design, implementation, testing and maintenance. The main functionality of the system is student registration, grade report, transcript generating, report generating, notice posting, attendance recording, timetable and exam schedule generating, discipline recording and resource managing. The project organized by the chapter. Each chapter contents the sub contents of the project. The chapters are Introduction of whole project process, Description of the Existing System, Proposed System, System Analysis, System Design, Implementation, Testing Plan and Conclusion. Thus, we need to develop an system which is a versatile and complete end-to-end school management software with precision engineered to enhance the administrative efficiency of Woodland University.

Overall, the project has helped us how to work in group with the help from our tutor Suresh Gautam.

I update the project is feature and welcome your comments and suggestions.

**Introduction**

     Almost every sector in this modern society has been computerized but the backbone of a nation’s development is mostly run by the clerical/ paper-based system. In every school or college, we can see some number of computers, but they are only used for learning. Using the computer is not only for learning, but they can also be used to manage the entire college from registration of students to printing the transcript of a student.

Course management software is the software that keeps the information secure and makes their management easier. This software provides the attractive environment where the data manipulation is quite advanced and information about the staff and students can be easily viewed when needed. The Course Management System is to manage the task related to the university students, staff and to reduce time in finding of appropriate member in collage view.

            Woodlands University College (WUC) is a small Higher Education institution which is offering a wide range of degree courses willing to change from clerical system to the modern computerized system. Currently the university is facing difficulties handling all the types of university works such as student, teacher and staff records along with the improper library record. They do not have proper software via which they can facilitate their students technically involving all the necessary things for a student and staff.

So, to overcome the problem they are facing they wanted new software which makes the job easier by saving all kinds of records easily and safely.

       The project is completed using Html, CSS, and JavaScript for the front-end development of the software and for the back-end development of the software we had used for PHP and Python.

**Project background**

          Woodlands University College (WUC) is a small Higher Education institution which is offering a wide range of degree courses where Dr. Simon White is currently a course leader. He wanted to change the system firstly to a pilot system then to the fully computerized system. They want the system to manage the record of all staff and students with the proper management of course and timetable. They also want the system to save the attendance with the facility of publishing the result using the software.

Some of the features which we included on the website:

1. Reduce complexity through automation. The automation of the system will help the organization in proper maintenance of the record, less manpower, less man-days, less cost, proper & accurate functioning
2. User-friendly environment to operate on.
3. Compatible to all platforms (like mobile and laptop).
4. Improve Administration/Accounts department’s efficiency
5. Backup and security
6. Enhance Users satisfaction and productivity
7. Flow of quick transition of information
8. Quick and fast service to the users.
9. In the clerical system, the data loss of the students/staff may occur, but the computerized system helps to make data safe and secure.

**Project Aims and Objectives**

**Project Aims**

* The proposed system is aimed to remove the drawbacks of the existing system and it to make a more secure system.
* As this system is fully computerized it will make the data management and other difficult task easy.
* This system will do every possible task that can be done using computers. Like creating, editing, updating, deleting, displaying the data.
* To store all the data electronically, this can be shared between the selected people or everyone.
* The system will not only for storing, it will be used to retrieve the data easily and will also be used for report generation.
* To reduce the working hours.
* To replace the paper based working methodology by fully functionalized computerized system

**Project Objectives**

The objective of this project is to develop GUI based software which is platform independent and user friendly. This will remove the data redundancy and will be fast in operation.

* Scheduling for evocation exercises.
* Scheduling for interview with clients.
* Prepare the interview questions for different clients. i.e. Admin, course leader, module leader, existing students
* To come across the problems of existing system that has been faced by the students staffs, management team.
* Preparing the appropriate map work of the beneficial website
* To find out the entities and attributes those are needed for the website work.
* Sketching the navigation diagrams, BON diagrams, mock up activity diagram.
* Making the ER diagrams
* Relating the ER diagram with proper attributes
* Implementing the code
* Comparing the system with other similar software.
* Presenting the software

**Project Development Methodology**

If client is satisfied

Requirement

Development

Testing

Maintaining

Deployment

**Requirement Engineering**

**2.1. Elicitation Activities**

**2.1.1.** **Interview Plans**

After the confirmation, interview was conducted to know what client want the system to function. Knowing what the client wants we designed the mockup until the client get satisfied after which we started the coding part which went through different testing until the system reply properly. Subsequently working system was handed to the client which will be maintained time to time.

1. Inauguration
2. Elicitation Activities
3. Design and Analysis
4. implementation
5. Testing and System Evaluation
6. Deployment
7. Maintenance
8. Inauguration: The document provided by the client helped us to configure what to do about the project. By going through the document, we were not sure what they wanted in the software. So, we decided to have an interview session with respective staff of the Woodlands University College.
9. Elicitation Activities: In the group, we have prepared some interview plans by making questions for the course leader, module leader, course administrator and existing student. After the interview session we have come up with the ideas and information which helps us to develop the software. We had also found out the problems which they faced while using the clerical based system by taking the interview session.
10. Design and Analysis: From the outcome of the interview session, we created a wireframe and mockup design if they were as for the client requirement. After the confirmation of the client, we started the development of the work.
11. Implementation: After getting the enough requirement and design document, the work is divided in different parts and coding is started which is main part and longest phase of the software development. The project is done by using PHP and MySQL in backend while html, CSS and JavaScript was used as frontend. While after implementation the end deliver is the product/ system.
12. Testing and System Evaluation: After the completion of development work, we were testing by assigning the username and password for the user. The work which should be done for the system was fully working. For the trial, we went to the course leader for the confirmation of functionality of the system.
13. Deployment: When the course leader evaluated the system and after agreement, we handed the system to the course leader.
14. Maintenance

After the deployment of software, it needs to be upgraded / change. The change will occur mostly in the content of module and different software operation which will be updated from time to time.

Team Decomposition

By collaborating with each of the group members, each part of the project’s tasks will be fulfilled. Each members of group have their own duty to successfully complete the project. The members of the group and the responsibilities specified in the table.

|  |  |  |
| --- | --- | --- |
| Number | Team member name | Responsibility |
| 1 (Leader) | Inshan K.C. | Coding, tester, and report |
| 2 | Pankaj Badu | Interview and backend |

Interview Title: School Management System Interview

Date: 2021-01-04

Person in attendance: • Inshan K.C.

• Pankaj Badu

• Ronak Bastola

The questions and respective answers for this interview are detailed in the table below:

|  |  |
| --- | --- |
| Problem domain entities | questions |
| Course leader | * For what reason are you implementing this website in your system? * Do you have any existing website? If yes, please describe about it. * If you want to replace your existing website then why do you want to replace it what failures were there in the previous website? * What difficulties did u face while using current website? * What types of content will you publish on the site? * What social media elements would you like integrated? * Do you need to integrate chat features? * Do you have any information regarding the security and policy? * Do you have any branding materials or styles sheets that you want to implement? * Do you like to add any special features to this website? * Any training needed for the use of this system? * What you want your new website to portray to ensure your faculty? * What you think about updating the course in the system? * Do you want layout grid or list view layout? * Shall system record student feedback? |

|  |  |
| --- | --- |
| Administrator | Questions |
|  | * The student must register to a course of his choice or they can see the full course detail before the registration? * What are the features you would like to have? * Do you want the system to notify for the confirmation of the student to the system? * Do you want the system to save the payment method and time? And notify the user when to pay and get salary. * Is your collage an independent body or part of an international franchises? * The system should be available during? * Do you want any feature that informs each student’s absences and his/her degrees in all subjects and issuing reports to student’s parents, with collected information whenever asked or needed? * How would you provide the notice regarding collage activities to the students? * Would you like to add digital attendance system? * How would you like to hear students problem through this system? * Would you like to add parents in the group so that u could contact with them? * Do you have a proposed sitemap prepared? * What kind of ads will you be running? * Will you require a responsive design? * What amounts do you like give for the system? * When should the prototype of the computerized system be finished? * For new students would you like to add informative page with admission details and contacts. |

|  |  |
| --- | --- |
| Module leader | questions |
|  | * Would you like to add class, add divisions, add subjects? * What feature do you like to add? * Do you want to Mark Attendance? * Do you want to send mock paper, check papers? * Do you want any feature where students can check their reports and grades? * Do you want to inform students about the timetable regarding your module? * How do you want to manage module contents? * Would you like to personalize content so that the content shown is targeted and relevant for different types of students? * Do you want the feature to create discussion forum by yourself? * Do you want any specific feature to communicate with PAT and students? * How can students submit their assignments through this system? * How students can put forward their quires regarding their problems? |

|  |  |
| --- | --- |
| Ex. Students | Questions |
|  | * How much does the current system affect the learning outcome of the students? * What problems did you face using this system? * How the assignment and results were given to the students in the current system? * How could you communicate with your module tutor for support? * How would you like to communicate with PAT? * Can a training session be beneficial for students? * How would you like to give feedbacks? * How would you like to submit your assignments? * Do you want any other informative sites to be linked with this system? |

Interview findings

This part contains the finding from the interview session.

Interview: With the Dr. Simon White

Interview Date: 2021-01-04

Duration: 44:27

Person in attendance:

Inshan KC

Pankaj Badu

Ronak Bastola

|  |  |  |
| --- | --- | --- |
| Interviewer | Question number | Question and client response |
| Pankaj Badu | 01 | For what reason are you implementing this website in your system?  Currently we are using manually system. So, I will not call it as 100% manual. We are using some kind of computerized system, but we are maintaining all the record in file and spreadsheet format. It is very difficult to make it public. Even if we share our file to the other user; when we put any other updates in the file it will stay in my system only. So that it will not be propagated and shared with somebody else. And again, we have to send the same document. It has been very difficult to manage all the records all the information is scattered in different system. So, it is been difficult to manage and access the latest information in day-to-day basis. |
| Pankaj Badu | 2 | Do you have any existing website? If yes, please describe about it.  Currently we are using excel spreadsheet file to record the attendance and any information in tabular format. We are using the word document to store the document based system which includes like paragraph, pictures, and so on. Normally we are using clerical system to maintain our current function |
| Pankaj Badu | 3 | If you want to replace your existing website then why do you want to replace it what failures were there in the previous website?  It has been really difficult in managing all types of record. Using the proposed developed site should help in the solving all the problem faced. We are facing the problem in preparing and uploading study material. Not only that we are also facing difficulty in tracking the student attendance. |
| Pankaj Badu | 4 | What difficulties did u face while using current website?  So, using the system, we are facing the difficulties in most of the aspect. First of all, I would like to mention about the difficulty in tracking the attendance and managing assignment along with the preparing and uploading the study material and keeping them up to date in one place is another challenge for us while using this system. This is also the problem for student. It is difficult for them to access the materials. We are sending the study material in the email. So, when they see the document, we may send them the same document multiple times. So, it is difficult for them to as well to track which is the latest one. |
| Pankaj Badu | 5 | What types of content will you publish on the site?  The content could be like in format of text, audio, picture or video file. I think these four types of material will be there which we have to manage in our learning management system. |
| Pankaj Badu | 6 | What social media elements would you like integrated?  Not really at this moment. We are not in this vision to in cooperate the any kind of social media in this system. We will think in future. |
| Pankaj Badu | 7 | Do you need to integrate chat features?  We will limit our communication in three mode: if users are aware of the name of person to whom they would like to ask the question then user should be able to ask in offline mode rather than online mode  If user would like to comminate in group with administrator or any staff  If user want to comment and features would be revealing the real identity of the commenter  If we have these kinds of communication mechanism, then it will be sufficient for now. Online chat features is not mandatory.  Normally I would prefer, email for receiving the assignment submission for tuor saying that “Hey \*\*\*\*\* \*\*\*\* has submitted the assignment” for events. The list of email should be available within the system so that tutor could see the notification which are dispatch for email login. These two kinds of mechanism will be sufficient. |
| Pankaj Badu | 8 | Do you have any information regarding the security and policy?  Yes. There should be role-based authentication. So authentication mechanism should be there which include like username and password. Then user should have access which will be given role. There are administrator role, teacher role, module leader, and student role. The role will include features that we are allowed to access in our system. |
| Pankaj Badu | 9 | Do you have any branding materials or styles sheets that you want to implement?  I did not have exact color thing and branding materials. You can use light black colored theme. You can put menu bar in left side which includes list of modules. When you click, you should view sub module and sub section in tree structure. After clicking tree structure, you should view respective pages and respective report in main section. |
| Pankaj Badu | 10 | Do you like to add any special features to this website?  I am good whatever you have asked and the requirement I have provided. This feature will be sufficient. |
| Pankaj Badu | 11 | Any training needed for the use of this system?  Definitely. It will be difficult to access and use the new system. First we need to conduct the training session and in addition to that we also have to prepare video demonstration of how to use system available in offline mode. |
| Pankaj Badu | 12 | What you want your new website to portray to ensure your faculty?  They should have access to send the mail to student and send the message to module leader. They can create the assignment along with the due date and reference material for the assignment. They also should have features to track your attendance and the time user have spent in the class when user login and logout. It should also track download the content and submitted assignment. The system should also publish the grade that should be visible only to the user. |
| Pankaj Badu | 13 | What you think about updating the course in the system?  The new course will be added by the administrator and they will assign the module leader to the course and once the course has got the syllabus and course materials then they will have further option to allocate tutors so that list of students will assign to the particular tutor and they can follow up activities like assigning course to particular student and student can have study material which are assigned to them. |
| Pankaj Badu | 14 | Do you want layout grid?  I prefer to put it in the user preference so that I can have option to choose whether I would like to see the content in a grid or list view. |
| Pankaj Badu | 15 | Shall system record student feedback?  Yes. Feedback and comment can be submitted in three method. The first method could be sent message or feedback to person pr in group like administrator or teacher group. Also, with comment without revealing the id. |

Interview Date: 2021-01-04

Interview: With the Mr. Adam Blake

Duration: 50:00

Person in attendance:

Inshan KC

Pankaj Badu

Ronak Bastola

|  |  |  |
| --- | --- | --- |
| Interviewer | Question number | Question and client response |
| Ronak Bastola | 01 | The student must register to a course of his choice or they can see the full course detail before the registration?  Student can enroll in the subject pr course once then they do the admission for particular course then administrator will approve and assign course to the student. So that the individual student will enroll in the particular course. |
| Ronak Bastola | 2 | What are the features you would like to have?  I would like to have complete administration in the system like in terms of course and module administration. So I should have the access of creating the module and course also with assigning the particular modules and courses to the teacher or student. Once the student is enrolled and course tutor and module leader are assigned to the course and module then students and tutors can do further interaction and create their class and start their class. Being the administrator I also want to see whether all the assignment and all the due date are properly followed and the attendance are properly maintained by the individual student so they can take part in final exam. |
| Ronak Bastola | 3 | Do you want the system to notify for the confirmation of the student to the system?  As soon as the event occur. Not only that if student have submitted and their tutor has finished their grading and publish their result and that publish result should be notified to student. So that the student can go to the portal and check their result and feedbacks provided by the tutor. |
| Ronak Bastola | 4 | Do you want the system to save the payment method and time? And notify the user when to pay and get salary.  That mechanism is not necessary in current version. Currently you can avoid this features. |
| Ronak Bastola | 5 | Is your collage an independent body or part of an international franchises?  Actually the college is partially independent. It means university will prepare the course syllabus and program. In terms of independency following the course syllabus approved by the university and they can take decision of how that have to conduct internal assignment and how many assignment to allocate to individual student and calendar they would like to follow. That kind of rules and regulations will be followed by on college administration rather than like dictating the university. College is partially dependent and independent. |
| Ronak Bastola | 6 | The system should be available during.  System should be available all the time. System should not limit in office hour. |
| Ronak Bastola | 7 | Do you want any feature that informs each student’s absences and his/her degrees in all subjects and issuing reports to student’s parents, with collected information whenever asked or needed?  Yes it is required. We have 80% as minimum requirement to sit or participate in the final exam which is mandatory. If we did not track the attendance in the system we will not implement that rule. At least 60% grading should have scored in each assignment which are allocated during semester. In case of losing or failing any kind of 60% score then that student will not have access to attempt final exam. For that we need to include such kind of features n the application. |
| Ronak Bastola | 8 | How would you provide the notice regarding collage activities to the students?  There will be dashboard where notification section should be available. Whatever notification publish to public or particular student should be visible in the dashboard. |
| Ronak Bastola | 9 | Would you like to add digital attendance system?  I would like to track attendance when they attend the class in online mode and also like to track how much they have spared within the class and also like to check how many times they have done the interaction within class. These three kinds of features I would like to track in online mode. |
| Ronak Bastola | 10 | How would you like to hear students problem through this system?  There are three comments and feedback mechanism. The first method could be sent message or feedback to person pr in group like administrator or teacher group. Also, with comment without revealing the id. |
| Ronak Bastola | 11 | Would you like to add parents in the group so that u could contact with them?  In this regard, its not mandatory to include parents in group emails. We can exclude this particular features. |
| Ronak Bastola | 12 | Do you have a proposed sitemap prepared?  I do not have particular requirement for this. You can put your navigation panel to left hand side. You can have list of module and sub module are available in tree structure. So that you can sort the page when clocked on menu. |
| Ronak Bastola | 13 | What kind of ads will you be running?  I do not need any kind of advertisement at this point. Instead, it should show the list of college affiliated to us and courses we are offering and list of services and program. |
| Ronak Bastola | 14 | Will you require a responsive design?  Yes, it is required. I should have option to view page in mobile, tablet and desktop version. |
| Ronak Bastola | 15 | What amounts do you like give for the system?  I would like to hear back from you in this regard. Youi can list down the features in which you can deliver the solution and put your price in budget amount. So that I will give you the green signal which option is suitable for me. |
| Ronak Bastola | 16 | When should the prototype of the computerized system be finished?  It should finish in hard deadline for 6 months. If you could develop the portal for 4 months and 1month for testing and rest for the documentation. |
| Ronak Bastola | 17 | For new students would you like to add informative page with admission details and contacts.  You can have notification page in student dashboard. So that student will ger aware of key information like admission date assignment due date grading published and so on. Once you have placed that features in dashboard that would be sufficient. |

Interview Date: 2021-01-04

Interview: With the Dr. Raj Singh

Duration: 30:00

Person in attendance:

Inshan KC

Pankaj Badu

Ronak Bastola

|  |  |  |
| --- | --- | --- |
| Interviewer | Question number | Question and client response |
| Inshan K.C. | 01 | Would you like to add class, add divisions, add subjects?  Normally classes and subject will be added by administrator. Being a module leader I would ask administrator for additional classes and subject if I have to. |
| Inshan K.C. | 2 | What are the features you would like to have?  Being a module leader, I would get assigned to particular module and verify the student material which are prepared and upload in the system and even I would communicate with the tutor who are going to teach the course in the individual classes. Also, I would like to track all the course progress and the schedule class with the assignment progress are on trac as per the plan as we have posted in the very beginning. I would like to do the sample for the assignment which are checked by the individual tutor and, I would like to check the attendance and level they are maintaining which ais the basic requirement I would like to see in this system. |
| Inshan K.C. | 3 | Do you want to Mark Attendance?  These facilities should be given by the system itself. |
| Inshan K.C. | 4 | Do you want to send mock paper, check papers?  I would like to avoid using my mail. So, system should have functionality to manage everything regarding mock test and examination. |
| Inshan K.C. | 5 | Do you want any feature where students can check their reports and grades?  As soon as the tutor will finish their grading then they should go for publishing result. Then individual student should have access toward their grade sheet. |
| Inshan K.C. | 6 | Do you want to inform students about the timetable regarding your module?  If we could have functionality to see the calendar in the dashboard and a when they have any kind of due dates then that should be published and visible in dashboard so that it will be easy for tracking to the student and even easy for the module leader as well as the tutor. |
| Inshan K.C. | 7 | How do you want to manage module contents?  Normally we will prepare the study material and upload in our learning management system. Being the module leader there should be functionality I should have using which I can make the content visible or invisible to the student or tutors. |
| Inshan K.C. | 8 | Would you like to personalize content so that the content shown is targeted and relevant for different types of students?  Currently not. You can put it in the future enhancement. |
| Inshan K.C. | 9 | Do you want the feature to create discussion forum by yourself?  I would like to have that feature included in my dashboard as well and can also be initiated by administrator. So this functionality should be accessible for both role. |
| Inshan K.C. | 10 | Do you want any specific feature to communicate with PAT and students?  I would like to avoid using emails and the chat conversation. I prefer to have offline conversation in which I will receive the offline questions and provide the response on it. Whatever the response I have provided could be copied in my email it would be awesome. |
| Inshan K.C. | 11 | How can students submit their assignments through this system?  Normally before submitting tutor will awssign the assignment to particular student. Individual student then will have access to that assignment and submit their report or document as an response to that assignment. |
| Inshan K.C. | 12 | How students can put forward their quires regarding their problems?  The first method could be sent message or feedback to person pr in group like administrator or teacher group. Also, with comment without revealing the id. |
| Inshan K.C. | 13 | How do you publish your result in current process?  Currently we are publishing the result in email. We want to replace that kind of mechanism. We want to publish result through system. Once result is published student can view the result. |

Interview Date: 2021-01-04

Interview: With the Mr. Mark Williams

Duration: around 15 minutes

Person in attendance:

Inshan KC

Pankaj Badu

Ronak Bastola

|  |  |  |
| --- | --- | --- |
| Interviewer | Question number | Question and client response |
| Inshan K.C. | 01 | How much does the current system affect the learning outcome of the students?  Its being very difficult. We are receiving all the materials in our email. We receive the course syllabus in an email and receive the learning material in our email. If some modification happens in our learning materials and that will be not reflected in our email. We again need to ask tutor to resend the study materials. We have seen the problem in assignment management. So, we receive the assignment question in our email which is maintained, and due date appears in text. It is very difficult to track. If we could have system having due date which will alert when we approach the due date. We are having trouble in attendance. Sometime our tutor will forgot mark our attendance in record. So, at the end of semesters, we will be in trouble if we do not have minimum attendance. Sometimes it is being very difficult when our tutor forgot to mark our attendance and our attendance record is very low.  For the attendance, it should be automatic. Tutor will create the schedule being the student is should login and access the class. As soon as I attend the class my attendance should be mark. And it should maintain how much time I spent time in class. So that it will be easy for our tutor to track our attendance as well the time spent in our class. |
| Pankaj Badu | 2 | What problems did you face using this system?  We are facing problem in attendance management system, learning materials and attendance. Everything is scattered so the information might be in other laptop or information might not being in up-to-date form. |
| Pankaj Badu | 3 | How were the assignment and results given to the students in the current system?  Currently we are receiving the feedback in the email. So, this is the processure we are having the practice in the current system.  Normally those record are confidential. As the record are sent to the respective email id still it is manual process which is tedious and time consuming. |
| Inshan K.C. | 4 | How could you communicate with your module tutor for support?  The communication medium should be in three form. |
| Pankaj Badu | 5 | How would you like to communicate with PAT?  The administrator will allocate personal tutor to student. The tutor can create a classroom and follow up with individual student.  Everything is manual and nothing is centralized and up to date information is rare. |
| Pankaj Badu | 6 | Can a training session be beneficial for students?  The student will start using new system then it will be issue for us and we may face lot of troubles. Before rolling out the new system its good to have a good training session. In addition to training session if there is a video tutorial posted in learning management system so that we can see video tutorial. |
| Pankaj Badu | 7 | How would you like to submit your assignments?  Before submitting the assignment, our module leader or course leader should assign the assignment to individual student. They should get notified through email and system. Once students finish their assignment, they should have option to submit through online portal. Also, should have option to see the grading when assignment is checked and the verified published result. |
| Pankaj Badu | 8 | Do you want any other informative sites to be linked with this system?  I would like to have the link of courser. |

**2.1.3** **Other problem domain research**

**2.1.3.1** **Comparable Software System review**

The comparable software system helps to know about the functionality and features used by the other pre-existing system which helps to make the software better. Some of the comparable software are listed below:

**2.1.3.1.1** **Coursera**

Coursera is a global online learning platform that offers anyone, anywhere, access to online courses and degrees from leading universities and companies. It helps to build the skills with the courses and provides certificate after complete ting the course. The degree is provided from the world-class universities.

Graphical user interface, text, application, website

Description automatically generated

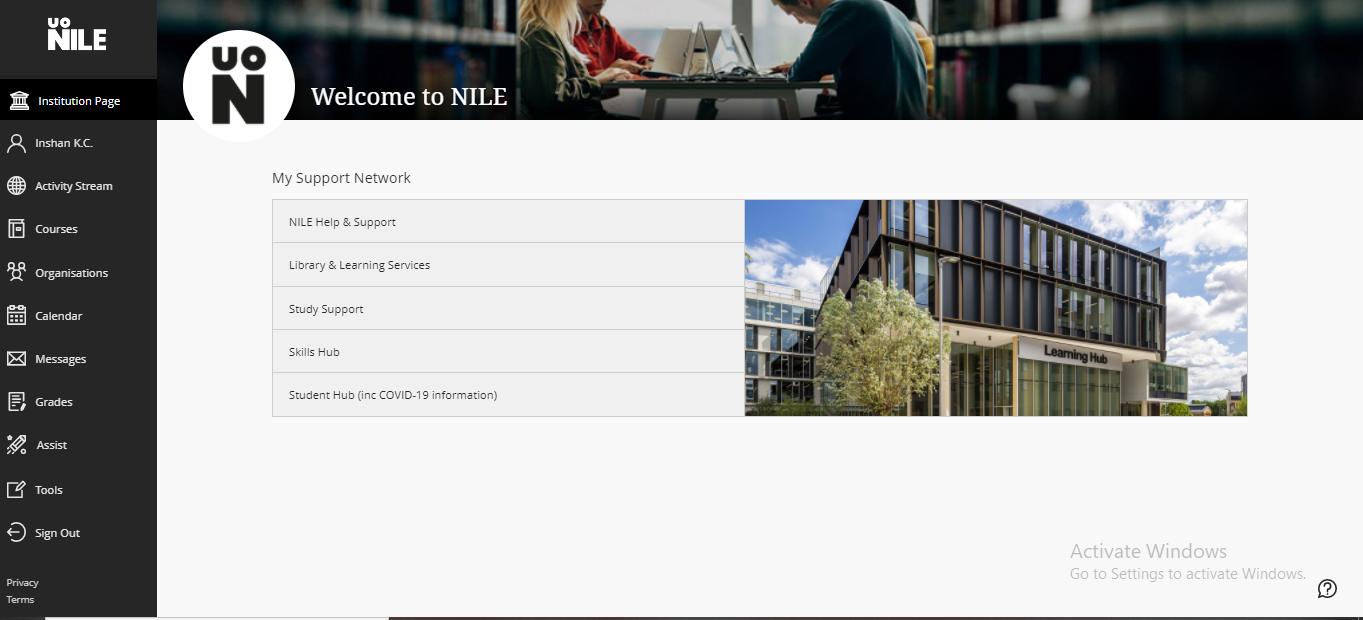
*Figure 1 : Coursera website*

|  |  |
| --- | --- |
| **Positive aspects** | **Negative aspects** |
| * **The page of coursera is responsive** * **UI is neat and clean** * **Coursera keeps track of information about course and the date the user has joined and deadline of the course**   **Joined** | * **Need training to be familiar with the system** * **UI is hard to understand** |

**2.1.3.1.3** **NILE**

It is a web-based software which helps to make the learning of the courses through online.

NILE is the online web software which manages the course content of university of Northampton. It has been used as an template for developing the site for woodland university college.



*Figure : NILE*

|  |  |
| --- | --- |
| **Positive aspects** | **Negative aspects** |
| * **the main functionality has been used as dashboard after login is completed** * **UI is neat and clean** * **User can update their own information and do not have to wait for the permission of the administrator** * **Can communicate with university, module leaders and other students using this software** * **User friendly** | * **Need training to be familiar with the system** * **All the functionality is in left bar which make the navigation bar too congested** |

2.1.3.2 **Development Relevant Legislation**

**Law and legislation**

**Data Protection Right**

The Data Protection Act 2018 controls how your personal information is used by organizations, businesses or the government. Everyone responsible for using personal data has to follow strict rules called ‘data protection principles’. They must make sure the information is: used fairly, lawfully and transparently. An Act to make provision for the regulation of the processing of information relating to individuals; to make provision in connection with the Information Commissioner’s functions under certain regulations relating to information; to make provision for a direct marketing code of practice; and for connected purposes.

The main principles of Data Protection Act are:

* Lawfulness, fairness and transparency.
* Purpose limitation.
* Data minimization.
* Accuracy.
* Storage limitation.
* integrity and confidentiality (security)
* Accountability.

**Digital Economy Act**

The act addresses policy issues related to electronic communications infrastructure and services, and updates the conditions for and sentencing of criminal copyright infringement. The Digital Economy Act 2017 (the Act) makes provision about electronic communications infrastructure and services, including the creation of a broadband Universal Service Order (USO), to give all premises in the UK a legal right to request a minimum standard of broadband connectivity.

This act is made up of 6 parts which are

* Access to digital services
* Digital infrastructure
* Online pornography
* Intellectual property
* Digital government
* Miscellaneous

**Computer Misuse Act**

The Computer Misuse Act protects personal data held by organizations from unauthorized access and modification). This refers to entering a computer system without permission (hacking).Unauthorized access to computer materials with intent to commit a further crime.

This act strictly prohibits the hacking process to use others data for the purpose of illegal works. The use of internet should be done within the limit of legislation and different illegal works like data theft, misuse data, online scam, and robbery should not be done.

**2.1.3.4** **Any other relevant problem domain investigation data**

The current global universities and colleges are facing the clerical or paper-based management system. The other relevant problem are covered below:

* In the clerical based system, the loss of the data is the major problem which causes critical affect to the universities.
* The record of the student data will be more complex in the paper-based management system.
* While publishing the result of the students, it will not be confidential way of publishing the result.
* The old-fashioned way of teaching the students make the learning outcome of the student poor.
* To communicate with the module tutor or with PAT is also difficult in the clerical based system.
* Hard to track the academic progress of the student.

**2.2** **Requirement Specification**

The application useful prerequisite and non-utilitarian necessity is talked about in this part and framework determination is made dependent on customer necessity.

**2.2.1** **Problem Domain Description**

The difficult space is a region of any issue that should be analyzed to take care of the issue. Here, to settle the business restrictions, the initial step is to do the means to decide all the issues. This segment decides and clarifies all the issues and restrictions in the current administrative arrangement of the college.

**2.2.1.1** **Existing business Operation**

Using clerical whole system of Woodland University is running. They are taking the admission of new student where they are registering, keeping their records to providing transcript using clerical system. Even the updated timetable has been provided using clerical.

User of current system

Users are the people who are part of Woodland University who comes directly interacted with the system. They perform the various task like registering, recording and maintaining timetable, generating reports, and so on.

|  |  |
| --- | --- |
| Users | Responsibility |
| Principal | Creating timetable, assigning classroom and PAT to the students, and managing the whole college |
| Record officer | Records student data, preparing grade report card, record teachers and student attendance |
| Staffs | manage discipline and resources/ properties |
| Teacher | Organizes student marks, teaches students, manages grade reports, and attends school |
| Student | attends class, get grade report card, view timetable, take grade report card, take exams |

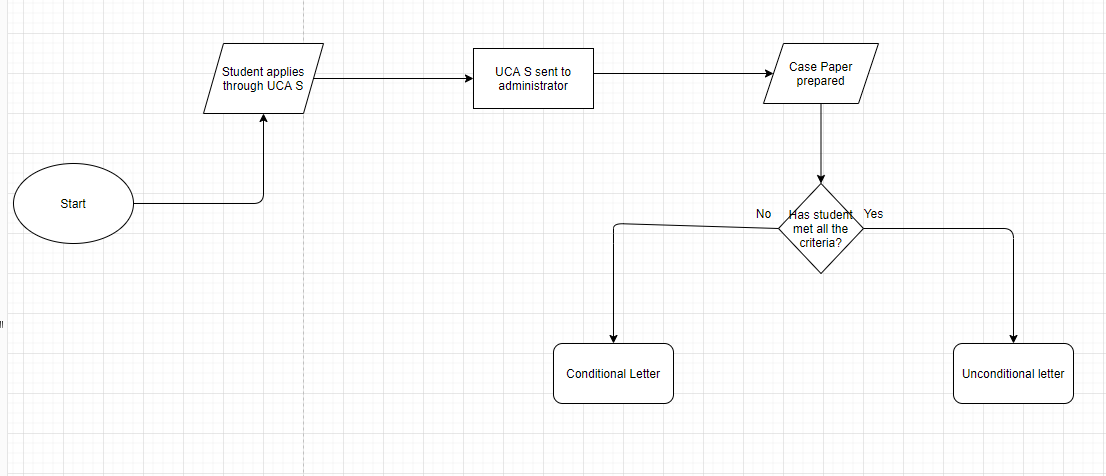
Diagram

Description automatically generated

As of now college is utilizing customary strategy for its everyday action which is paper based. A paper-based framework is the most established framework to store information in it. The sponsorships structures are additionally in paper-based structures.

2.2.1.1.1. UCAS and letter

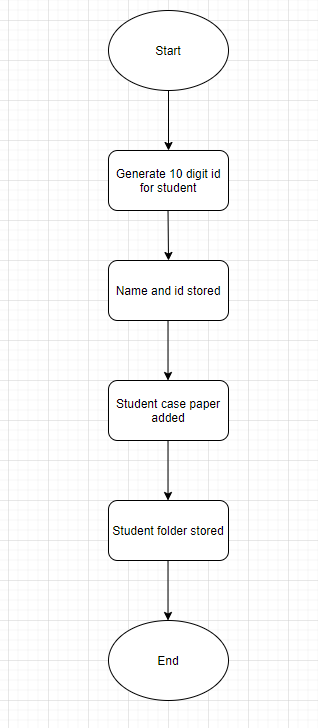
When a student applies to the university through UCAS, it sends all the document files which are necessary to the university to intake him/her. When the check process is end and the documents are checked if they met all the criteria unconditional letter is sent to the student if not then conditional letter is sent.



**Fig: Showing the ongoing process of admission UCAS**

**2.2.1.1.2.** **Id generation and student folder**

When a student is enrolled to the university. He/she is given an id number. A different database is made for the student where all the information of the student is kept. The database folder of the student is kept along with the case paper.

****

**Fig: Showing the generation of id and storage**

**2.2.1.2** **Summary of existing business limitations**

All the cycles recommend the accompanying results that should be settled:

* The affirmation measure is a long cycle.
* Since the records are put away, they are powerless against being damaged by fire or some other sort of harm.
* Since the declarations are actually put of some place in the notification sheets, there is a high possibility of passing up the declarations.
* There is no confirmation of protection of information.
* Accessing and update of the record in whole administration is time taking
* Redundancy check. Multiple records for the same person.
* Any people can access the record of the student. With the system t=only authorized people can access the file
* Inadequate space for storing ever increasing files.
* Generating the transcript of the student is difficult.
* Current system do not produce the required the report to allows parent to view the status of their current children.

Bottleneck problem of the existing system

Performance: As the system is manual, it takes time for school management to be carried out, such as report generation, student information and grade management, and timetable and schedule creation.

Input and output

Input

* Data is not captured in time to be useful.
* Data is not correctly captured; errors are included.
* Data is collected redundantly, and the same information is collected more than once.

Output

* Lack necessary and relevant information.
* Sometimes Information is not in a useful format like registering user information.
* Information is not timely to its subsequent use.

Security and control

* Too little security or control
* Input data is not adequately edited because of using manual system.
* Data damage with like dust or rain so they are storing in room and shelf even on working desk.
* Redundantly stored data is inconsistent in different files.
* Data privacy regulations or guidelines can be violated.
* Processing errors are occurring by people.
* Decision- making errors are occurring.

Efficiency

Waste of time

Data is redundantly input or copied.

Data is redundantly processed.

Information is redundantly generated.

Waste of materials and suppliers.

Effort required for tasks is excessive.

Materials required for tasks are excessive

**2.2.2.** **Functional Requirements**

This part covers the necessary functionalities of the proposed framework. The functionalities proposed should tackle all the current issues in the college’s present framework. This bit of the report gives an outline about the entrance levels which are administrator, module pioneer, understudies, and their jobs/advantages in the framework.

**2.2.2.1** **Required Features of the system**

As all the current business restrictions have been distinguished, the accompanying highlights are viewed as fitting answers for the issues.

**2.2.2.1.1.** **Database and backup**

As the current framework is an administrative based framework, paper the executives has been frenzied and dull. To beat this issue the proposed framework will totally store all the vital data and information in the data set which can be effortlessly got to at some random time. The issue of arranging all the lethargic records following 10 years of capacity will be totally gone as the information base will clutch the information in the framework. This will likewise address the issues of absence of extra rooms in the actual setting.

**2.2.2.1.2.** **Students record**

Since the confirmation cycle of the current framework has a progression of steps which end up being extensive and wasteful, the framework will control all the 3 phases/status of the understudies dependent on their connection with the school/course. The 3 specific statuses which will be managed are temporary, live and lethargic. Each understudy either right now enlisted or dropped or the ones that simply applied to the college will have a specific status and their information will be taken care of in like manner. Just the live understudies will approach all data gave by the framework.

**2.2.2.1.3.** **Staff records**the framework will hold and deal with all the information identified with the staffs that have ever been related to the college. The concerned specialists will have the option to add, alter or erase the data identified with the staffs and the staffs will hold certain functionalities as indicated by their assignment.

**2.2.2.1.4.** **Course records**

* The directors can add/alter/erase a course.
* An understudy should be taken a crack at a specific course.
* Modules, levels and term should be allocated to a course.

**2.2.2.1.5.** **Assignment records**

* The module pioneers will have the option to add/alter/erase task content.
* Module pioneers can add the accommodation date and accommodation link.
* The understudies will have the option to see and present a task.

**2.2.2.1.6.** **Attendance management**

* The module pioneers can add the participation data of any understudy.
* For a low participation, the understudies are sent an admonition message/email.
* The live understudy’s data will likewise incorporate the participation rate.

**2.2.2.7.** **Personal Tutor management**

* An individual guide is relegated to each understudy.
* The PAT and the understudies will have the option to impart in the framework.

**2.2.2.8.** **Grades management**

* The module pioneers will have the option to review the tasks of the understudies.
* The evaluations will either be temporary or last.
* The module pioneers can add criticisms/remarks alongside their evaluations.
* The understudies will have the option to see their evaluations of every one of their module.

**2.2.2.9.** **Announcement management**

* The announcement can be added by the chairmen.
* The announcement for modules can be added by module pioneers.
* Module announcement can be seen simply by the module chief and the understudies.

**Access rights/ level of access**

2.2.2.10 LOGIN

Since the system needs to handle a lot of confidential student information, a login function is essential for ensuring security. Users who do not have the correct access rights will be prevented from connecting to the database. There are three groups of users of the system with different access rights: ¬ Principal – acts as the system administrator and can perform all functions, ¬ Staff – can view, input, and modify records of all students and teachers. ¬ Teachers – can view, input, and modify information of the students they teach. This function also allows users to change their own password.

2.2.2.11 Student Information Management

This function allows the personal information of students, such as their name, sex address, etc., to be managed. Using this function, a user can: ¬ Insert a new student record. ¬ Make changes to the student information. ¬ Remove a student record if the student graduates or leaves the school. ¬ Search for a particular student or group of students.

2.2.2.12 Student Performance Management

This function allows the user to maintain the details of the academic performance of the students, such as the examination results, their conduct, etc. Using this function, a user can: ¬ Insert a subject and its paper. ¬ Modify a subject’s details (e.g., name, weighting, etc.) ¬ Insert examination marks for a student. ¬ Make changes to the examination marks of a student. ¬ Insert a conduct evaluation for a student. ¬ Make changes to the conduct evaluation of a student. ¬ Search for a particular student or group of students.

2.2.2.13 Student Extracurricular Activity Management

As each student is encouraged to join a large variety of extracurricular activities, the system will handle this information for the students. Using this function, a user can: ¬ Insert a new activity. ¬ Modify an activity’s details. ¬ Indicate that a student participates in an activity. ¬ Make changes to the activity records of a student. ¬ Insert a new award. ¬ Modify an award’s details. ¬ Indicate that an award was won by a student. ¬ Make changes to the awards of a student. ¬ Search for a particular student or group of students. 2.2.2.14 Staff Information Management: The school needs to change the records of teachers, such as the particulars, their qualifications, the classes they teach, etc. Using this function, a user can: ¬ Insert a new record for a teacher. ¬ Make changes to the record of a teacher. ¬ Search for a particular teacher or a group of teachers in a class.

2.2.2.14 Classroom Management: In addition to the information about students; the school also needs to store the information for different classes for effective school management. This information includes the class mistress, the teachers for each subject of the class, classroom allocation for each lesson, etc. Using this function, a user can: ¬ Insert a new class. ¬ Make changes to the class details. ¬ Assign students to a new class. ¬ Assign subjects taken by a student. ¬ Assign teachers to teach a subject for a class. ¬ Generate timetables. ¬ Search for a particular data or group of data in a class or a group of classes. 2.2.2.15 Graduate Information Management: When a student graduates or leaves the school, for the sake of future references, the records of the student should be kept, presumably, forever. To manage the data of the graduates, this function allows a user to: ¬ Move a student’s record to the graduate database when she graduates or leaves the school. ¬ Search for information of a graduate or group of graduates. ¬ Search on the performance of a graduate or group of graduates. ¬ Search on the extracurricular activities and the awards of a graduate.

2.2.2.16 Report Generation: The school needs to produce many kinds of reports and documents throughout the whole academic year. This function provides the necessary paper output, such as records printing, report printing, etc. Some standardized documents, such as the class lists, timetables, etc. for teachers, students and parents can also be printed.

2.2.2.17 Registration

To stores the necessary information of those actors of school management.

**2.2.1.10 Degree of Access**

This bit gives a review pretty much all the advantages given to various client gatherings. A few functionalities are regular to the client gatherings while some shift. The managers, module pioneers and understudies all have various jobs and are given various functionalities as indicated by the need.

Administrators record management

|  |  |  |  |
| --- | --- | --- | --- |
| Operation | Administration | Module leader | Student |
| Add administrator | Add the new administrator | No access | No access |
| Edit administrator | Change the administrator’s information | No access | No access |
| Delete administrator | Delete the administrator’s record completely | No access | No access |
| View administrator | Can view all kind of record | No access | No access |
| Archive administrator | Archive teacher | No access | No access |

|  |  |  |
| --- | --- | --- |
| Name of entity | Datatype | Notes |
| User id | INT (15) | Primary key for administrator records table |
| User first name | VARCHAR2 (25) | First name of the administrator |
| User middle name | VARCHAR2 (25) | middle name of the administrator |
| User last name | VARCHAR2 (25) | last name of the administrator |
| Date of Birth | DATE | Birth date of a user |
| Contact | VARCHAR2 (15) | Contact of user |
| Gender | Enum (male, female, other) | Gender of the user |
| Email | VARCHAR2 (25) | Email of the user |

Teacher record management

|  |  |  |  |
| --- | --- | --- | --- |
| Operation | Administration | Module leader | Student |
| Add teacher | Add the new hired teacher | No access | No access |
| Edit teacher | Change the teacher information | No access | No access |
| Delete teacher | Delete the teacher record completely | Delete the published assignment | No access |
| View teacher | Can view all teacher record | Can view all student record who are taking the course | Can view their own profile but not others |
| Archive teacher | Archive teacher | No access | No access |

|  |  |  |
| --- | --- | --- |
| Name of entity | Datatype | Notes |
| teacher id | INT (15) | Primary key for teacher record table |
| course id | INT (10) | foreign key from content table |
| Year | DATE | Year in which teacher joined woodland university |
| Level | INT (8) | Level in which they are currently teaching |
| course name | VARCHAR2(19) | Name of the course which they are teaching currently |
| experience | Int (5) | Year of experience in teaching sector |
| Student id | INT (15) | Id of assigned student for pat meeting |

Student record management

|  |  |  |  |
| --- | --- | --- | --- |
| Operation | Administration | Module leader | Student |
| Add student | Add the new student each year | No access | No access |
| Edit student | Change the student information | No access | No access |
| Delete student | Delete the student record completely | No access | No access |
| View student | Can view all student record | Can view all student record who are taking the course | Can view their own profile but not others |
| Archive student | Archive student | No access | No access |

|  |  |  |
| --- | --- | --- |
| Name of entity | Datatype | Notes |
| Student id | INT (15) | Primary key for student record table |
| course id | INT (10) | foreign key from content table |
| Year | DATE | Year in which student joined woodland university |
| Level | INT (8) | Level in which they are currently studying |
| course name | VARCHAR2(19) | Name of the course which they are currently taking |
| GPA | VARCHAR2(5) | Grade they have scored |
| Teacher id | INT (15) | Foreign key from teacher table |
| Personal tutor id | INT (15) | Id of assigned personal tutor |

Course Management

|  |  |  |  |
| --- | --- | --- | --- |
| Operation | Administration | Module leader | Student |
| Add course | Add the new course | No access | No access |
| Edit course | Change the course each year | No access | No access |
| Delete course | Delete the module completely | No access | No access |
| View course | View the course | View the course | View the course in which they are enrolled |
| assign course | Assign the teacher what they will be studying | View what they will be teaching | View what they will be reading |
| Archive course | Archive content | No access | No access |

|  |  |  |
| --- | --- | --- |
| Name of entity | Datatype | Notes |
| course id | INT (10) | Primary key for content table |
| Year | DATE | Year in which they are studying the course |
| Date | DATE | Date in which course has been updated |
| course name | VARCHAR2(19) | Name of the course |
| Description | VARCHAR2(255) | Description of the course |
| Teacher id | INT (15) | Foreign key from teacher table |

Module Management

|  |  |  |  |
| --- | --- | --- | --- |
| Operation | Administration | Module leader | Student |
| Add module | Add the new module in the course | No access | No access |
| Edit module | Change the module each year | No access | No access |
| Delete module | Delete the module completely | No access | No access |
| View module | View the module | View the content of the module for each term | View the module which they will be studying each year |
| assign module | Assign the teacher what they will be studying | View what they will be teaching | View who their module leader are |
| Archive module | Archive content | No access | No access |

|  |  |  |
| --- | --- | --- |
| Name of entity | Datatype | Notes |
| module id | INT (10) | Primary key for content table |
| Year | DATE | Year in which they are studying the module |
| Title | VARCHAR2(50) | Title of the module |
| Module id | INT (10) | Foreign key from module table |
| Date | DATE | Date the module has been updated |
| Module name | VARCHAR2(19) | Name of the module which course has been provided |
| Description | VARCHAR2(255) | Description of the module |
| Teacher id | INT (15) | Foreign key from teacher table |

Content Management

|  |  |  |  |
| --- | --- | --- | --- |
| Operation | Administration | Module leader | Student |
| Add content | Add the new content in the module | No access | No access |
| Edit content | Change the content of module as necessary | No access | No access |
| Delete content | Delete the content of a module completely | No access | No access |
| View content | View the content of the course or module | View the content of the module for each term | View the content of the module for each term |
| Download content | Download the content of the module for each term | Download the content of the module for each term | Download the content of the module for each term |
| Archive content | Archive content | No access | No access |

|  |  |  |
| --- | --- | --- |
| Name of entity | Datatype | Notes |
| content id | INT (10, no decimal place) | Primary key for content table |
| term id | INT (15) | Foreign key |
| Title | VARCHAR2(50) | Title of the given content |
| Module id | INT (10) | Foreign key from module table |
| Date | DATE | Date the assignment has been updated |
| Module name | VARCHAR2(19) | Name of the module which course has been provided |
| Description | VARCHAR2(255) | Description of the content |

Assignment Management

|  |  |  |  |
| --- | --- | --- | --- |
| Operation | Administration | Module leader | Student |
| Add assignment | No access | Add assignment for a student | No access |
| Edit assignment | No access | Change the assignment | No access |
| Delete assignment | No access | Delete the published assignment | No access |
| View assignment | View the assignment | View the assignment | View the assignment of the module for each term |
| Submit Assignment | No access | No access | Only student can submit the assignment |
| Archive assignment | No access | Archive grade | No access |

|  |  |  |
| --- | --- | --- |
| Name of entity | Datatype | Notes |
| Assignment id | INT (10, no decimal place) | Primary key for assignment table |
| Student id | INT (15) | Foreign key from student table |
| Title | VARCHAR2(50) | Title of the given assignment |
| Module id | INT (10) | Foreign key from module table |
| Date | DATE | Date the assignment has been published |
| Module name | VARCHAR2(19) | Name of the module which grade has been provided |
| Description | VARCHAR2(255) | Description of the assignment |

Grade Management

|  |  |  |  |
| --- | --- | --- | --- |
| Operation | Administration | Module leader | Student |
| Add grade | Add attendance | Add attendance link for a student | No access |
| Edit grade | Change the grade | Change the grade of a student | No access |
| Delete grade | Delete the grade | Delete the grade | No access |
| View grade | View the grade of a student | View the grade of a student | View the grade of the module for each term |
| Archive grade | Archive grade | Archive grade | No access |

|  |  |  |
| --- | --- | --- |
| Name of entity | Datatype | Notes |
| Grade id | INT (12) | Primary key for grade table |
| Student id | INT (15) | Foreign key from student table |
| Teacher id | INT (15) | Foreign key from teacher table |
| Module id | INT (10) | Foreign key from module table |
| Date | DATE | Date the grade has been published |
| Module name | VARCHAR2(19) | Name of the module which grade has been provided |
| Assignment id | INT (10, no decimal place) | Foreign key from assignment |

Attendance Management

|  |  |  |  |
| --- | --- | --- | --- |
| Operation | Administration | Module leader | Student |
| Add attendance | Add attendance | Add attendance link for a student | No access |
| Edit attendance | Change the attendance | Change the attendance of a student | No access |
| Delete attendance | Delete the attendance | Delete the attendance link | No access |
| warning | Send an email to parent and student of low attendance | Send an email to parent and student of low attendance | No access (view) |
| View attendance | View the attendance of a student or teacher | View the attendance of a student | View the attendance |
| Archive attendance | Archive attendance | Archive attendance | No access |

|  |  |  |
| --- | --- | --- |
| Name of entity | Datatype | Notes |
| Attendance id | INT (12) | Primary key for attendance table |
| Student id | INT (15) | Foreign key from student table |
| Teacher id | INT (15) | Foreign key from teacher table |
| Total days | INT (5) | Total number of running days of university |
| Present days | INT (5) | Total days in which student is present |
| Absent days | INT (5) | Total days in which student is absent |
| Date | DATE | Date in which attendance is taken |

Personal Tutor Management

|  |  |  |  |
| --- | --- | --- | --- |
| Operation | Administration | Module leader | Student |
| Assign personal tutor | Assign a student with a teacher | No access (view) | No access (view) |
| Edit P A T | Change the personal tutor for any student | No access | No access |
| Delete P A T | Delete the personal tutor for any student | No access | No access |
| Archive P A T | Archive personal tutor and student | No access | No access |
| View P A T | Update personal tutor and student | View who they are assigned | View who their P A T ais |
| Edit P A T meeting time | Change meeting time between student and their assigned P A T | View | View |

|  |  |  |
| --- | --- | --- |
| Name of entity | Datatype | Notes |
| pat\_id | INT (12) | Primary key for PAT table |
| Student id | INT (15) | Foreign key from student table |
| Teacher id | INT (15) | Foreign key from teacher table |
| Room no | INT (5) | Room in which pat meeting is held |
| Start time | DATE | Time in which meeting starts |
| End time | DATE | Time in which meeting ends |

Low Attendance, Grades published, Assignment published, Assignment submitted, Pat meeting schedule

Dear XYZ,

If this is the first time that you have enrolled at the woodland, you will need to set up a secure account to allow you to access University campus-based computers and online services which includes NILE, our on-line learning environment. You have been given a **temporary password** which **you MUST change.**

If **you have attended** a face-to face enrolment session **at the University,** you should have changed and checked your temporary password with a member of staff as part of the enrolment process. You should not need to do anything further.

If you **have not attended** a face-to-face enrolment session, or have not had the opportunity to change and check your temporary password, you need to do the following:

1. You must login to the user portal page to change your password,
2. You must enter your Student ID as your Username name. **Your Student ID is: XXXX**
3. You must enter your temporary password. This is made up of:
   * 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 Farmer and your date of birth is 8 October 1986, your password would be: 081086Fa=

1. To keep your account secure you will be asked to set up security questions.
2. Click Change Password and set a **new password** as instructed.
3. Next, click on My Profile. You should see a **four-digit PIN.** This PIN is used in conjunction with your student ID for printing and to access your online library account. If you wish to change your PIN, click on Edit Your Information, enter a new four-digit PIN and click Save Changes.
4. You can now access the University Student Hub which includes NILE our on-line learning environment.

If you have any issues with completing the above, please ring XXX XXXX XXXX XXX XXX.

Grade Published

|  |
| --- |
| **Date: XXXX/XX/XX**  **Dear: X Y Z,**  **Announcements**  **Your result of term 1 of XXX XXXX tca/ assignment has been published. Please check your result.**  **Best regards,**  **Woodland University College** |
| |  | | --- | |  | |

Assignment published

|  |
| --- |
| **Date: XXXX/XX/XX**  **Dear: X Y Z,**  **Announcements**  **The assignment of term 1 of XXX XXXX has been published. Please check your announcement.**  **Best regards,**  **Woodland University College** |
| |  | | --- | |  | |

On TCA

|  |
| --- |
| **Date: XXXX/XX/XX**  **Dear: X Y Z,**  **TCA exam**  **Your TCA exam of term 1 of XXX XXXX tca has been published. Please check the date of TCA and be prepared for the TCA. Good luck.**  **Best regards,**  **Woodland University College** |
| |  | | --- | |  | |

PAT meeting

|  |
| --- |
| **Date: XXXX/XX/XX**  **Dear: X Y Z,**  **PAT meeting**  **I have scheduled a meeting for XXXX-XX-XX at time in room number. Please be on time.**  **Best regards,**  **PAT name** |
| |  | | --- | |  | |

**2.2.3** **Performance Requirements/ Non-Functional Requirements**

This part of the report gives a thought regarding the speed and nature of the different functionalities of the framework. It decides how rapidly and how dependable the finished result framework will be.

**2.2.3.1** **Speed**

* Throughput: The throughput of the information will be more prominent and bringing the records will be a smooth cycle.
* Response time: The objective reaction time for the framework will associate with 4-6.5 seconds making it generally somewhat higher than normal reaction time. In any case, around 10 seconds of normal speed would by actualized.

**2.2.3.2** **Capacity**

The limit of the framework is focused on at first to be around 2000 individuals. In any event 2000 individuals will have the option to get to the framework simultaneously without having any slack in the reaction. The measure of information and records to be put away at first is focused to be around 60,000. Anyway, the limit can generally be disclosed by the need.

**2.2.3.3** **Reliability**

The framework plan (UI) will be of a simple and reasonable level. Anybody can figure out how to utilize the framework and discovering functionalities won’t be troublesome as the route board will give pretty much every area for the client to get the essential use out of the framework.

2.2.3.4 Security

System should be used only by allowed users and users can use the system as per given rights.

2.2.3.5 Reliability

System should be designed in such a way that system should not be affected by system failure or any errors occurred during execution.

2.2.3.6 Portability

It should be Platform Independent.

2.2.3.7 Performance

System should be robust and fast for any number of users.

2.2.3.8 User friendliness

The package developed is easy to learn and understand. Even a new user can use the system effectively, without any difficulty. The help and user manuals are provided to solve the further queries of the users. With the help of the user manuals the user can get the full details of the functionality of the system.

2.2.3.9 User satisfaction

The package is such that it stands up to the user’s expectations. The system is successful in generating the reports of the task status and details of the student. The package is currently being successfully run in the organization.

2.2.3.10 Error handling

Responses to user’s errors and undesired situation have been taken care of to ensure that the system operates without halting. Proper error handling codes are put with the codes. Security and robustness: The package are able to avoid or tackle disastrous action. It allows only the authentic user to access the software as it is protected by the username and the password. All the administrative tasks can the project leader only, so the illegal intervention is not possible. 2.2.3.11 Modularity

The package has relatively independent and single function parts that are put together to make complete system. Thus, because of this modular approach the system, despite being robust is not cumbersome. It also contributes for the fast execution of the system.

2.2.3.12 Maintainability

The system can decrease the time and effort for program maintenance. The full details of the projects being undertaken, and the task status corresponding to each student is stored carefully and the reports are generated as per the requirement.

2.2.3.13 Timeliness

The package can operate well under normal peak and recovery conditions.

2.2.3.14 Availability

The system doing 24 hours in every day unless there is no internet connection as per client request.

**2.2.4** **Design Constraints**

This segment clarifies the front end non-useful necessities that influence the framework. It clarifies how the framework is assembled not what the framework does. The limitation have been forced by the customer himself and each imperative has been thought about for building the framework.

* **Target Operating System:**

The framework configuration will be viable with each leading operating framework in the market which are Windows, Linux, Android, and so on.

* **Distributed or neighborhood design:**

The engineering will be neighborhood to the college including additional capacity for reinforcements.

* **Required equipment necessities:**

The framework will have the option to work in any given equipment either PCs or cell phones.

* **Front end realistic styles:**

The customer has not given a particular shading inclination at the same time, the framework will follow the shading blend utilized in the logo of the college to look after consistency.

* **Programming Language:**

There are no programming language imperative by the customer side, anyway the customer has mentioned to utilize an item situated methodology (OOP) in this way, and the group has chosen to utilize PHP as the base language for building the framework. However, the code structure will carefully actualize the MVC design for a deliberate turn of events.

* **Design approaches:**

The customer has mentioned to utilize Object Oriented Design approach for the advancement of the framework so OOP will be carefully executed.

2.2.5 **Commercial Constraints:**

As we have isolated the undertaking into four-stage and designated season of about a month. The plan requires seven days including a wide range of wireframes and model plan. Improvement requires fourteen days which incorporates both front-end, back-end, and information base worker arrangement for both web applications and programming. Testing requires 10 days which likewise goes close by with the last phase of the improvement of the product. Finally, organization requires 3 days of work which incorporates all the cycles like getting information to the data set, establishment of the product to the framework, and facilitating the web application.

|  |  |  |  |
| --- | --- | --- | --- |
| Project Information | | | |
| Total length of the project | | 20 weeks | |
| Hourly wage: | | $100 | |
| Number of workers: | | 3 | |
| Project Breakdown | | | |
| Sections | Week | Hours per week for each person | Total wage |
| Requirement specification | 2 | 5 | $3000 |
| Design and analysis | 5 | 6 | $9000 |
| Project build breakdown | 12 |  |  |
| Console | 20% | 10 | $7200 |
| Website | 40% | 15 | $21600 |
| Mobile application | 15% | 12 | $6480 |
| Testing | 22% | 12 | $6336 |
| Evaluation | 3% | 5 | $540 |
|  |  |  |  |
| Maintenance | The cost will be further be negotiated | | |
|  |  |  |  |
| Additional Cost | | | |
| Reason | | Cost | |
| Software/ hardware | | $4550 | |
| Office costs | | $1525 | |
| Total building cost | |  | |
| Other expenses | | $2000 | |
| Profit margins | | $8550 | |
| Program rights | | All rights to college administrator | |
|  | |  | |
|  | |  | |
| Total project cost | | $70781 | |

|  |  |  |  |
| --- | --- | --- | --- |
| Project Information | | | |
| Total length of the project | | 40 weeks | |
| Hourly wage: | | $50 | |
| Number of workers: | | 3 | |
| Project Breakdown | | | |
| Sections | Week | Hours per week for each person | Total wage |
| Requirement specification | 4 | 3 | $1800 |
| Design and analysis | 10 | 4 | $6000 |
| Project build breakdown | 24 |  |  |
| Console | 20% | 5 | $3600 |
| Website | 40% | 9 | $12960 |
| Mobile application | 15% | 4 | $2160 |
| Testing | 22% | 8 | $6336 |
| Evaluation | 3% | 2 | $216 |
|  |  |  |  |
| Maintenance | The cost will be further be negotiated | | |
|  |  |  |  |
| Additional Cost | | | |
| Reason | | Cost | |
| Software/ hardware | | $4550 | |
| Office costs | | $3000 | |
| Total building cost | |  | |
| Other expenses | | $2000 | |
| Profit margins | | $5550 | |
| Program rights | | All rights to college administrator | |
|  | |  | |
|  | |  | |
| Total project cost | | $38372 | |

|  |  |  |  |
| --- | --- | --- | --- |
| Project Information | | | |
| Total length of the project | | 52weeks | |
| Hourly wage: | | $30 | |
| Number of workers: | | 3 | |
| Project Breakdown | | | |
| Sections | Week | Hours per week for each person | Total wage |
| Requirement specification | 2 | 5 | $900 |
| Design and analysis | 10 | 2 | $1800 |
| Project build breakdown | 40 |  |  |
| Console | 20% | 3 | $4320 |
| Website | 40% | 4 | $5760 |
| Mobile application | 15% | 6 | $3240 |
| Testing | 22% | 4 | $3168 |
| Evaluation | 3% | 2 | $216 |
|  |  |  |  |
| Maintenance | The cost will be further be negotiated | | |
|  |  |  |  |
| Additional Cost | | | |
| Reason | | Cost | |
| Software/ hardware | | $4550 | |
| Office costs | | $5000 | |
| Total building cost | |  | |
| Other expenses | | $2000 | |
| Profit margins | | $2550 | |
| Program rights | | All rights to college administrator | |
|  | |  | |
|  | |  | |
| Total project cost | | $29404 | |

Hardware requirement

Processor: Pentium IV, 1.2GHz or better processor

RAM: 512 MB or better RAM

Monitor: 9” or bigger

Hard disk capacity: 16GB or better space

Keyboard: Standard

Mouse: Any mouse

Printer: Any printer

Web connectivity: yes

Software requirement

Operating system: Windows 2000 or above

Front end: HTML, JavaScript, and CSS

Back end: MYSQL and PHP

Web browser: any

**3.** **System design and analysis**

System analysis is a management technique which helps us in designing a new system or improving an existing system.

The analysis part in any system design is the most important process to achieve a better understanding in the requirement of different functionalities for the system that is being designed. 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 for objects
* Interaction of the objects
* Operation of the objects
* The necessary internal procedures carried out by the objects

**3.1.** **Preliminary design stages**

To analyze the given in scenario in a better way to make it more efficient, many processes are carried out. Many techniques are used to identify the potential classes which are within the system. Below mentioned techniques are considered model for the OPP approach as the use of different processes of techniques are used to give the most favorable result for class identification.

**3.1.1.** **Textual analysis**

This part of analysis is the used to identify the candidates of class. This method of analysis shows us that noun is corresponding to any candidate class and verb is corresponding to the responsibilities obsessed by the class.

Through the principles of this method, we could identify total 4 classes which are admin, course leader, module leader, students.

|  |  |
| --- | --- |
| **Candidate class** | **Responsibilities of candidate class** |
| **Admin** | Add admin, edit admin, update admin, delete admin, archive admin, assign course leader, Add course leader, edit course leader, update course leader, delete course leader, archive course leader, assign module leader, Add module leader, edit module leader, update module leader, delete module leader, archive module leader, assign students, update student information, add announcements, add PAT, allocate PAT time |
| **Course leader** | assign courses, update course information |
| **Module leaders** | Add module contents, edit module contents, delete module contents, add assignment contents, edit assignment contents, grade assignments, add announcements, add attendance sheets |
| **Students** | View module contents, view assignments, view grades, view task, view grades, view attendance, view assignment files |
| **courses** | Add course, edit course, delete course, view course, update course, archive course |
| **modules** | Add modules, edit module, delete module, update module, view module, archive module |
| **Assignments** | Add assignments, delete assignments, update assignments, assign assignments, display assignments, archive assignments |
| **Grades** | Add grades, display grades |
| **Announcements** | Add announcements, delete announcements |
| **Accounts** | add account, edit account contents, view contents |
| **Report** | Select, view report, save report |
| **Time table** | Add time-table, edit time-table, delete time-table,  View time-table |

**3.2.2.** **Significant event analysis**

Event analysis of the system includes the mapping of the events with the classes and the candidate attribute of the performers. Mostly, consideration of the events helps in the system design flow, that leads to a efficient mapping of the event flow of the software.

These events are considered to be candidate class routines, performance includes the candidate system class and attribute contains candidate attributes for the candidate system classes.

|  |  |  |
| --- | --- | --- |
| **Events** | **Performers** | **Candidate attributes** |
| **login** | * Admin * Module leader * students | * user id * password |
| **Sign up** | * Admin * Module leader * students | * user id * password * first name * middle name * last name * gender * date of birth * contact * address(permanent/temporary) * email address * course id * module id * level * blood group(amendatory) * usertype |
| **Logout** | * admin * module leader * students | * user id * password * first name * middle name * last name * gender * date of birth * contact * address(permanent/temporary) * email address * course id * module id * level * blood group(amendatory) |
| **Add/edit/delete**  **announcement** | * Admin * Module leader | * password * first name * middle name * last name * gender * date of birth * contact * address(permanent/temporary) * email address * module id(module leader) |
| **Add/edit/delete admin** | * Admin | * Admin id * Password * First name * Middle name * Last name * Gender * Date of birth * Contact * address(permanent/temporary) * email address * course id |
| **Add/edit/delete**  **Module leader** | * Admin | * Admin id * Password * First name * Middle name * Last name * Gender * Date of birth * Contact * Address * Email id * Course id * Module id |
| **Add/edit/delete**  **Students** | * Admin | * Admin id * Password * First time * Middle name * Last name * Gender * Date of birth * Contact * Email address * Student id * Course id |
| **Add course** | * Admin | * Admin id * Password * First time * Middle name * Last name * Gender * Date of birth * Contact * Email address * Course id |
| **Add modules** | * Admin | * Admin id * Password * First time * Middle name * Last name * Gender * Date of birth * Contact * Email address * Module id |
| **Add/edit/delete**  **Module content** | * Module leader | * Admin id * Password * First time * Middle name * Last name * Gender * Date of birth * Contact * Email address * Course id * Module id |
| **Add/edit/delete assignments** | * Module leader | * Module leader id * Password * First time * Middle name * Last name * Gender * Date of birth * Contact * Email address * Course id * Module id * Assignment id |
| **Grade assignment** | * Module leader | * Module leader id * Password * First name * Middle name * Last name * Gender * Date of birth * Contact * Email address * Course id * Module id * Assignment id * Student id |
| **Add/edit/delete attendance** | * Module leader | * Module leader id * Password * First name * Middle name * Last name * Gender * Date of birth * Contact * Email address * Course id * Module id * Month\_id * Total days * Present days * Absent days * Student id |
| **View assignments** | student | * Student id * Password * First name * Middle name * Last name * Gender * Date of birth * Contact * Email address * Course id * Module id * Level |
| **Request for information change** | student | * Module leader id * Password * First name * Middle name * Last name * Gender * Date of birth * Contact * Email address * Course id * Module id * Level |
| **View module content** | student | * Student id * Password * First name * Middle name * Last name * Gender * Date of birth * Contact * Email address * Course id * Module id * Level |
| **View assignments** | student | * Student id * Password * First name * Middle name * Last name * Gender * Date of birth * Contact * Email address * Course id * Module id * Assignment id * Level |
| class | Teacher  student | * Class id * Class name * Faculty id * Class span |
| EVENT | Teacher  student | * Event id * Venue * Date * Event name |
| feedback | student | * feedbackid * feedback * date * status * email |
| PARENT detail |  | * Username * Parent id * Children id * Address * Contact no * Dob * Occupation * Blood group * Qualification |

**3.1.3** **Commands Queries and Constraints**

The CQC technique is proficient way of filtering the classes produced by the significant event analysis. For every identified class, a series of evaluation is theoretically to done under the specific heading commands, queries and constraints. Command spell out the responsibility of the class, Queries spell out the attributes and constraints spell out the logical rules by which the class must line up with.

|  |  |  |
| --- | --- | --- |
| **Class** | **Module** | **Part: 1/1** |
| **TYPE OF OBJECT**  Add, edit, delete and view module content |  | |
| **Queries** | * Module leader id * Password * First name * Middle name * Last name * Gender * Date of birth * Contact * Email address * Course id * Module id * Assignment id | |
| **Commands** | * Add assignments * Edit assignments * Delete assignments | |
| **Constraints** | * Each assignment must have a brief * Each assignment must have a submission date * Each produced assignment can be edited * Each produced assignment can be deleted | |

**3.2** **Detailed Static System Design**

In this part, we have clear knowledge about the system requirements and all the entities for the systems are found out. By using static system design approach, the components and design for the system are completed. Some of the approaches used for static modeling are: BON diagrams, Use Case diagrams and Class diagrams. All these approaches give the basic idea about working for the system which helps us to make it easier for the implementation process.

**3.2.1.** **First Draft BON system Architecture Diagram**

**3.2.1.1. BON Cluster Charts**

|  |  |  |  |
| --- | --- | --- | --- |
| **CLUSTER** | User | | **Part:1/1** |
| **PURPOSE:** co-operating the structure of the system with the authorization | | **INDEXING:**  **Author:** Everest | |
| **Class** | | **Description** | |
| Admin | | Admin act as the main role model for the use of the system, who has all the authorities for the modification, deleting, updating, editing the records of staffs, students, and updating the notices. | |
| Staff | | Staff keeps the records regarding the students like student management, updates the timetable, manages the courses, assignment management | |
| Student | | Active role of the university/collage | |
| Ex-student | | They can view their past assignments, study materials | |

|  |  |  |  |
| --- | --- | --- | --- |
| **CLUSTER** | User Interface | | **Part:1/1** |
| **PURPOSE:** after the user enters to the interface, user can login to the system | | **INDEXING:**  **Author:** Everest | |
| **Class/(Cluster)** | | **Description** | |
| Academic website | | In this page all the records of university are kept | |
| Login | | Where user can get access to the system | |
| Navigation bar | | User can select can any of the given the options as per his/her requirement after they login to the system | |

|  |  |  |  |
| --- | --- | --- | --- |
| **CLUSTER** | User Database | | **Part:1/1** |
| **PURPOSE:** providing all the records to the users involved in the academic activities | | **INDEXING**  **Author:** Everest | |
| **Class/(Cluster)** | | **Description** | |
| Staff records | | All records regarding the staff academic performances | |
| Students records | | Complete records of student | |
| Course records | | Complete information about the course content | |
| Attendance records | | Complete record of the attendance of students and staff. | |
| Ex- students records | | Information about the Ex-students | |

|  |  |  |  |
| --- | --- | --- | --- |
| **CLUSTER** | Announcement | | **Part:1/1** |
| **PURPOSE:** updating students about the academic activities | | **INDEXING:**  **Author:** Everest | |
| **Class/(Cluster)** | | **Description** | |
| Assignment / TCA/Mock-exam | | Provided by the module leaders. | |
| Attendance | | The notification of student and staffs attendance records | |
| Grades | | Marking for the submitted assignments/TCA/Mock-exam | |
| Academic functions | | The notifications related to the course contents, academic activities and other extra activities performed by the university. | |

|  |  |  |  |
| --- | --- | --- | --- |
| **CLUSTER** | Reports | | **Part:1/1** |
| **PURPOSE:** complete information of the students enrolled to the university | | **INDEXING:**  **Author:** Everest | |
| **Class/(Cluster)** | | **Description** | |
| Exam | | Record of the exam performed and attended students | |
| Attendance | | The record of student attendance | |

|  |  |  |  |
| --- | --- | --- | --- |
| **CLUSTER** | News | | **Part:1/1** |
| **PURPOSE:** notice for the selected students | | **INDEXING:**  **Author:** Everest | |
| **Class/(Cluster)** | | **Description** | |
| Feedback | | Record about the students behavior and academic activities | |

**3.2.1.2 BON class charts**

|  |  |  |  |
| --- | --- | --- | --- |
| **Class** | Admin | | **Part:1/1** |
| **Type of object:** admin in the organization | | **INDEXING:**  Author:Everestt | |
| **Queries** | | Admin\_id, admin\_full name, admin\_password | |
| **Commands** | | Create, Amend, records, Display, Assign, edit, Search, update. | |
| **Constraints** | | * Queries cannot be null. * They must be enrolled to the university * Id number is given at creation, id numbers are never changed | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Class** | Students | | **Part:1/1** |
| **Type of object:** student enroll on the course | | **INDEXING:**  Author: Everest | |
| **Queries** | | Student\_id, student\_password, student full name, student\_DOB, student\_address, student\_contact, student photo, student\_level, email, gender, blood group | |
| **Commands** | | Create, Amend, archive, Display | |
| **Constraints** | | * Student\_id, first Name, middle, last name, address and Contact number cannot be null. * Must be enrolled to the university * Id number is given at creation, numbers are never changed. * Must be assigned to at least one course. | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Class** | Staff | | **Part:1/1** |
| **Type of object:** staff of the organization | | **INDEXING:**  Author: Everest | |
| **Queries** | | Staff\_id, staff\_name, staff\_middlename,  Staff\_lastname, staff\_dob, staff\_address, staff\_contact, staff photo, staff\_gender, joinedDate, qualification, staff\_photo. | |
| **Commands** | | Create, Amend, Archive, Display, | |
| **Constraints** | | * Staff\_id, first name, middle name, last name, Address, job title and Contact number cannot be null. * Must be enrolled to the university * Id number is given at the creation, numbers. Numbers are not changed * Students must be assigned to them * They must join at least one course | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Class** | Courses | | **Part:1/1** |
| **Type of object:** courses of university | | **INDEXING:**  Author: Everest | |
| **Queries** | | course\_id, admin\_id, staff\_id, course\_name, course\_leader | |
| **Commands** | | Create, Amend, Archive, Display, Delete, assign | |
| **Constraints** | | * The id for the course should be unique and not null. * Students cannot be enrolled with more than one course. * Tutor should be assigned to course. * admin\_id and staff\_id as foreign key. * A course leader should be appointed. | |

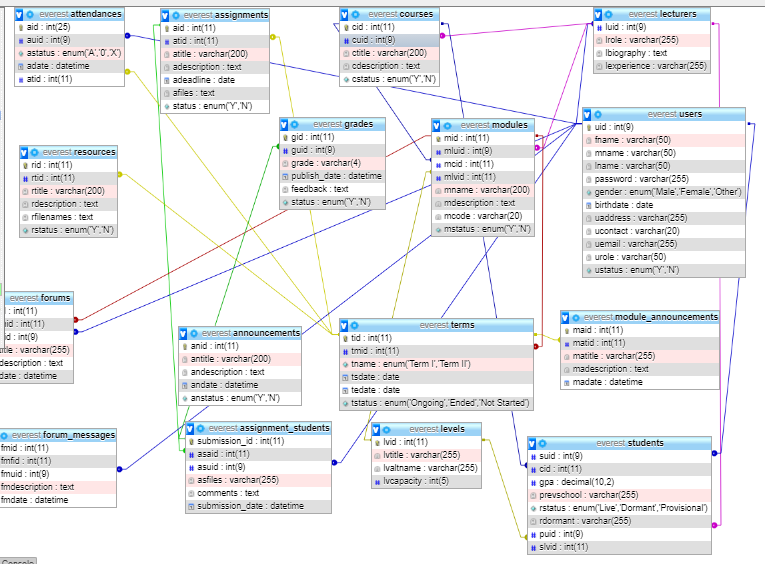
|  |  |  |  |
| --- | --- | --- | --- |
| **Class** | Modules | | **Part:1/1** |
| **Type of object:** module in the organization | | **INDEXING:**  Author: Everest | |
| **Queries** | | admin\_id, staff\_id, Module\_id , module\_name, module\_leader, course\_id | |
| **Commands** | | Create, Amend, Archive, Display, Delete, edit | |
| **Constraints** | | * Module id must be unique. * A module is assigned with at least one or more than one courses. * Module leader must be appointed. * Staff\_id, admin\_id, course\_id act as foreign key. | |

* + 1. **Detailed Static System Designs:**
    2. **First Draft BON System Architecture Diagram**

**Diagram

Description automatically generated**

Fig showing BON diagram



**Fig er diagram**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Datatype** | **Constraint** |
| users | | |
| uid | INT(9) | PK |
| fname | VARCHAR2(50) |  |
| mname | VARCHAR2(50) |  |
| lname | VARCHAR2(50) |  |
| password | VARCHAR2(255) |  |
| gender | ENUM(“male”, “female”, “other”) |  |
| Birthdate | DATE |  |
| uaddress | VARCHAR2(255) |  |
| ucontact | VARCHAR2(20) |  |
| uemail | VARCHAR2(255) |  |
| urole | VARCHAR2(50) |  |
| usatus | ENUM(“Y”, “N”) |  |

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Datatype** | **Constraint** |
| grades | | |
| gid | INT(11) | PK |
| guid | INT(9) |  |
| grade | VARCHAR2(4) |  |
| Pubish\_date | DATE TIME |  |
| feedback | TEXT |  |
| status | ENUM(“Y”, “N”) |  |

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Datatype** | **Constraint** |
| assignment\_students | | |
| Submission\_id | INT(11) | PK |
| asaid | INT(11) |  |
| asuid | INT(9) |  |
| asfiles | VARCHAR2(255) |  |
| comment | TEXT |  |
| submission\_date | ENUM(“Y”, “N”) |  |

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Datatype** | **Constraint** |
| courses | | |
| cid | INT(11) | PK |
| cuid | INT(9) |  |
| ctitle | VARCHAR2(255) |  |
| cdescription | TEXT |  |
| cstatus | TEXT |  |

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Datatype** | **Constraint** |
| announcements | | |
| anid | INT(11) | PK |
| antitle | VARCHAR2(255) |  |
| andescription | TEXT |  |
| andate | DATE TIME |  |
| anstatus | ENUM(“Y”, “N”) |  |

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Datatype** | **Constraint** |
| module\_announcement | | |
| maid | INT(11) | PK |
| matid | INT(11) |  |
| matitle | VARCHAR2(255) |  |
| madescription | TEXT |  |
| madate | DATE TIME |  |

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Datatype** | **Constraint** |
| Forum\_message | | |
| fmid | INT(11) | PK |
| fmftid | INT(11) |  |
| fmuid | INT(9) |  |
| fmdescription | TEXT |  |
| fmdate | DATE TIME |  |

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Datatype** | **Constraint** |
| forum | | |
| fid | INT(11) | PK |
| fmtid | INT(11) |  |
| fuid | INT(9) |  |
| fdescription | TEXT |  |
| fdate | DATE TIME |  |

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Datatype** | **Constraint** |
| levels | | |
| lvid | INT(11) | PK |
| lvtitle | VARCHAR(255) |  |
| lvalname | VARCHAR(255) |  |
| lvcapacity | INT(5) |  |

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Datatype** | **Constraint** |
| students | | |
| suid | INT(9) | PK |
| cid | VARCHAR2(50) |  |
| gpa | DECIMAL(10,2) |  |
| prevschool | VARCHAR2(50) |  |
| rstatus | VARCHAR2(255) |  |
| rdormant | ENUM(‘LIVE’,’DORMANT’, ‘PROVISIONAL’) |  |
| puid | INT(9) |  |
| slvid | INT(11) |  |

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Datatype** | **Constraint** |
| lectures | | |
| luid | INT(9) | PK |
| lrole | VARCHAR2(50) |  |
| lbiography | TEXT |  |
| lexoerience | VARCHAR2(50) |  |

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Datatype** | **Constraint** |
| terms | | |
| tid | INT(11) | PK |
| tmid | INT(11) |  |
| tname | ENUM(‘TERM I’, ‘TERM II’) |  |
| tsdate | DATE |  |
| tedate | DATE |  |
| tstatus | ENUM(‘ONGOING’, ENDED, ‘NOT STARTED’) |  |

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Datatype** | **Constraint** |
| assignments | | |
| aid | INT(25) | PK |
| atid | INT(11) | FK |
| atitle | VARCHAR(200) |  |
| adescription | TEXT |  |
| adeadline | DATE |  |
| afiles | TEXT |  |
| status | ENUM(‘Y’, ‘N’) |  |

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Datatype** | **Constraint** |
| Resources | | |
| Rid | INT(11) | PK |
| rtid | INT(11) |  |
| rtitle | VARCHAR(200) |  |
| rdescription | TEXT |  |
| rfilenames | TEXT |  |
| rstatus | ENUM(‘Y’, ‘N’) |  |

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Datatype** | **Constraint** |
| Resources | | |
| Rid | INT(11) | PK |
| rtid | INT(11) |  |
| rtitle | VARCHAR(200) |  |
| rdescription | TEXT |  |
| rfilenames | TEXT |  |
| rstatus | ENUM(‘Y’, ‘N’) |  |

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Datatype** | **Constraint** |
| modules | | |
| mid | INT(11) | PK |
| mluid | INT(9) |  |
| mcid | INT(11) |  |
| mlvid | INT(11) |  |
| mname | VARCHAR(200) |  |
| mdescription | TEXT |  |
| module | VARCHAR(20) |  |
| mstatus | ENUM(‘Y’, ‘N’) |  |

**Use Case diagram**

**It is also known as the blueprint of designation of system. A use case diagram is a representation of how user interacts with a system and what the system replies to the user. It is used to show how different kind of user (teacher, admin, and student) portray system and how they can use and what features they can use /assigned in the system. Using the use case diagram, it provides a graphical view of what system can do. It helps in giving higher view level of system to users. It is an outline of the system which can provide a interpretation ow the system is going to be designed.**

**Users initialize/request which acts as a primary actor while the server acts as a secondary actor which gives a response to actor. An arrow is connecting to a use case where arrow denotes user is using the section in system. In this system several scenarios, each representing different input request the server where it provides the output. The box represents the system.**

**Diagram

Description automatically generated**

**Fig use case diagram for the admin**

**Diagram

Description automatically generated**

**Fig use case diagram for teacher**

**Diagram

Description automatically generated**

**Fig use case diagram for student**

Use case documention

|  |  |
| --- | --- |
| Use Case Title | Registration |
| Use Case ID | 1 |
| Description: To create a new use and store the information in the database | |
| Pre-condition: admin should be login to system. Username should not be similar with another user. | |
| Task sequence  1 Normal Flow   |  |  |  |  | | --- | --- | --- | --- | | S.N. | User action | S.N. | System response | | 1 | Admin goes to login page | 2 | System open the login page. | | 3 | Admin enters their respective emails and password. |  | If username and password is correct, then system is redirected to home page/ dashboard. | | 4 | Admin then enters on system. |  |  | | 5 | Admin clicks registration link | 6 | System represents registration type (teacher or student) | | 7 | Admin selects registration type | 8 | System display the form which needed to be filled to create an account for student or teacher | | 9 | Admin fills the form and click register button | 10 | System verify the information | | 11 | Admin confirms the filled form | 12 | System confirms the form and store the data of users in database. |   2 Alternative flow  If there is error while filling the data on the system (10) which verifying system return the step to 6 in which admin must fill the form again by correcting error by giving appropriate data in the system. | |
| Post Conditions: User registered message will be seen. | |
| Primary actor: Admin | |
| Author: Everest | |

|  |  |
| --- | --- |
| Use Case Title | Log in |
| Use Case ID | 2 |
| Description: To use system an authorized user should log in using their assigned username and password. | |
| Pre-condition: all the user should use appropriate username and password and have an account. | |
| Task sequence  1 Normal flow:   |  |  |  |  | | --- | --- | --- | --- | | S.N. | User action | S.N. | System response | | 1 | Users go to login page | 2 | System open the login page. | | 3 | User enters their respective emails and password. (For new users, they must use their assigned password and username.) | 5 | If username and password is correct, then system is redirected to home page/ dashboard. | | 4 | Users then enter on system. |  |  |   2 Alternative Flow   |  |  |  |  | | --- | --- | --- | --- | | S.N. | User action (if failed) | S.N. | System response | |  |  | 5.1 | If error occurs, system will display error message (username or password is invalid). And system returns the step to 3 where user must fill the login form again. | | 5.2 | Users try again to login |  |  | | 5.3 | User fills the form or chooses the forget password. | 5.4 | System redirects page to forget password | | 5.5 | User fills necessary information to change password. |  |  | | 5.6 | Users then enter/click on change password button. |  |  | |  |  | 5.7 | System check user information from database and update their password and returns to step 3 where user can use their changed password. If any error is faced system return to step 5.5. | | |
| Post Conditions: If the ID or password is correct the user must be logged in. | |
| Primary Actor: Admin, Teacher, and Student | |
| Author: Everest | |

|  |  |
| --- | --- |
| Use Case Title | Logout |
| Use Case ID | 3 |
| Description: to logout from system. | |
| Pre-condition: Login to the system. | |
| Task sequence  1 Normal flow   |  |  |  |  | | --- | --- | --- | --- | | S.N. | User action | S.N. | System response | | 1 | Users click logout button | 2 | System closes the dashboard and open the login page. |   2 Alternative flow   |  |  |  |  | | --- | --- | --- | --- | | S.N. | User action | S.N. | System response | |  |  | 2.1 | Check the network if user is not logged out and user should logout again. | | |
| Post Conditions: User is logged out from the system. | |
| Primary Actor: Admin, Teacher, and Student | |
| Notes: User should not forget logging out from system after logging in. | |
| Author: Everest | |

|  |  |
| --- | --- |
| Use Case Title | Timetable |
| Use Case ID | 4 |
| Description: To create and view timetable | |
| Pre-condition: Admin should be logged in | |
| Task sequence  1 Normal flow:   |  |  |  |  | | --- | --- | --- | --- | | S.N. | User action | S.N. | System response | | 1 | Admin clicks on timetable management. | 2 | System displays option. | | 3 | Users select create option. | 4 | System displays form. | | 5 | Admin fills the form for creating timetable. | 6 | System verify each time and request for confirmation. | | 7 | Admin click confirm button. | 8 | System stores time and display. | | 9 | Admin click view timetable from timetable management. | 10 | System displays timetable created by admin. | | 11 | Admin choose delete option. | 12 | System requests confirm. | | 13 | Admin selects confirm and delete timetable. | 14 | System delete timetable from database. |   2 Alternative Flow   |  |  |  |  | | --- | --- | --- | --- | | S.N. | User action (if failed) | S.N. | System response | |  |  | 8.1 and  14.1 | If system does not create and delete timetable, check if network is available or not. After connection, try again. | |  |  | 10.1 | If there is no list, system display error message. | | |
| Post Conditions: User can view timetable. | |
| Primary Actor: Admin | |
| Author: Everest | |

|  |  |
| --- | --- |
| Use Case Title | Teacher |
| Use Case ID | 7 |
| Description: To display teacher details | |
| Pre-condition: Admin should be logged in | |
| Task sequence  1 Normal flow:   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | S.N. | | User action | | S.N. | System response | | | Create | | | | | | | | 1 | | Admin clicks teacher management button. | | 2 | System display the options. | | | 3 | | Admin select create option. | | 4 | System display form. | | | 5 | | Admin fills the form and click create button. | | 6 | System verify the form and store the data in database. | | | Update/ view | | | | | | | | 7 | | Admin clicks teacher management button. | | 8 | System display the options. | | | 9 | | Admin selects update option. | | 10 | System display the data | | | 11 | | Admin edits and clicks update button. | | 12 | System verify the filled form and request for confirmation. | | | 13 | | Admin confirms. | | 14 | System updates the data and store in database. | | | 15 | | Users enter teacher id and click search button. | | 16 | System verifies teacher id and displays teacher information. | | | Delete | | | | | | | | 17 | Admin clicks teacher management button. | | 18 | | | System show the options. | | 19 | Admin selects delete option. | | 20 | | | System request for confirmation. | | 21 | Admin confirms to delete. | | 22 | | | System delete data from list and database. |   2 Alternative Flow   |  |  |  |  | | --- | --- | --- | --- | | S.N. | User action (if failed) | S.N. | System response | |  |  | 4.1 | If invalid teacher id is given system displays an error message and return to step 3 | |  |  | 2.1 and  4.1 | If system does not create and updates exam schedule, check if network is available or not. After connection, try again. | | |
| Post Conditions: User can view student information. | |
| Primary Actor: Admin | |
| Author: Everest | |

|  |  |
| --- | --- |
| Use Case Title | Student |
| Use Case ID | 8 |
| Description: To display teacher details | |
| Pre-condition: Admin should be logged in | |
| Task sequence  1 Normal flow:   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | S.N. | | User action | | S.N. | System response | | | Create | | | | | | | | 1 | | Admin clicks student management button. | | 2 | System display the options. | | | 3 | | Admin select create option. | | 4 | System display form. | | | 5 | | Admin fills the form and click create button. | | 6 | System verify the form and store the data in database. | | | Update/ view | | | | | | | | 7 | | Admin clicks student management button. | | 8 | System display the options. | | | 9 | | Admin selects update option. | | 10 | System display the data | | | 11 | | Admin edits and clicks update button. | | 12 | System verify the filled form and request for confirmation. | | | 13 | | Admin confirms. | | 14 | System updates the data and store in database. | | | 15 | | Users enter student id and click search button. | | 16 | System verifies student id and displays student information. | | | Delete | | | | | | | | 17 | Admin clicks student management button. | | 18 | | | System show the options. | | 19 | Admin selects delete option. | | 20 | | | System request for confirmation. | | 21 | Admin confirms to delete. | | 22 | | | System delete data from list and database. |   2 Alternative Flow   |  |  |  |  | | --- | --- | --- | --- | | S.N. | User action (if failed) | S.N. | System response | |  |  | 4.1 | If invalid student id is given system displays an error message and return to step 3 | |  |  | 2.1 and  4.1 | If system does not create and updates student id, check if network is available or not. After connection, try again. | | |
| Post Conditions: User can view student information. | |
| Primary Actor: Admin | |

|  |  |
| --- | --- |
| Use Case Title | Timetable |
| Use Case ID | 7 |
| Description: To display teacher details | |
| Pre-condition: Admin should be logged in | |
| Task sequence ­­  1 Normal flow:   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | S.N. | | User action | | S.N. | System response | | | Create | | | | | | | | 1 | | Admin clicks timetable management button. | | 2 | System display the options. | | | 3 | | Admin select create option. | | 4 | System display form. | | | 5 | | Admin fills the form and click create button. | | 6 | System verify the form and store the data in database. | | | Update/ view | | | | | | | | 7 | | Admin clicks timetable management button. | | 8 | System display the options. | | | 9 | | Admin selects update option. | | 10 | System display the data | | | 11 | | Admin edits and clicks update button. | | 12 | System verify the filled form and request for confirmation. | | | 13 | | Admin confirms. | | 14 | System updates the data and store in database. | | | Delete | | | | | | | | 17 | Admin clicks timetable management button. | | 18 | | | System show the options. | | 19 | Admin selects delete option. | | 20 | | | System request for confirmation. | | 21 | Admin confirms to delete. | | 22 | | | System delete data from list and database. |   2 Alternative Flow   |  |  |  |  | | --- | --- | --- | --- | | S.N. | User action (if failed) | S.N. | System response | |  |  | 4.1 | If invalid teacher id is given system displays an error message and return to step 3 | |  |  | 2.1 and  4.1 | If system does not create and updates exam schedule, check if network is available or not. After connection, try again. | | |
| Post Conditions: User can view student information. | |
| Primary Actor: Admin | |
| Author: Everest | |

|  |  |
| --- | --- |
| Use Case Title | Exam schedule |
| Use Case ID | 7 |
| Description: To display teacher details | |
| Pre-condition: Admin should be logged in | |
| Task sequence ­­  1 Normal flow:   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | S.N. | | User action | | S.N. | System response | | | Create | | | | | | | | 1 | | Admin clicks schedule management button. | | 2 | System display the options. | | | 3 | | Admin select create option. | | 4 | System display form. | | | 5 | | Admin fills the form and click create button. | | 6 | System verify the form and store the data in database. | | | Update/ view | | | | | | | | 7 | | Admin clicks schedule management button. | | 8 | System display the options. | | | 9 | | Admin selects update option. | | 10 | System display the data | | | 11 | | Admin edits and clicks update button. | | 12 | System verify the filled form and request for confirmation. | | | 13 | | Admin confirms. | | 14 | System updates the data and store in database. | | | Delete | | | | | | | | 17 | Admin clicks schedule management button. | | 18 | | | System show the options. | | 19 | Admin selects delete option. | | 20 | | | System request for confirmation. | | 21 | Admin confirms to delete. | | 22 | | | System delete data from list and database. |   2 Alternative Flow   |  |  |  |  | | --- | --- | --- | --- | | S.N. | User action (if failed) | S.N. | System response | |  |  | 4.1 | If invalid teacher id is given system displays an error message and return to step 3 | |  |  | 2.1 and  4.1 | If system does not create and updates exam schedule, check if network is available or not. After connection, try again. | | |
| Post Conditions: User can view student information. | |
| Primary Actor: Admin | |
| Author: Everest | |

|  |  |
| --- | --- |
| Use Case Title | Pat |
| Use Case ID | 7 |
| Description: To display pat for student | |
| Pre-condition: Admin should be logged in | |
| Task sequence ­­  1 Normal flow:   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | S.N. | | User action | | S.N. | System response | | | Create | | | | | | | | 1 | | Admin clicks pat management button. | | 2 | System display the options. | | | 3 | | Admin select create option. | | 4 | System display form. | | | 5 | | Admin fills the form and click create button. | | 6 | System verify the form and store the data in database. | | | Update/ view | | | | | | | | 7 | | Admin clicks pat management button. | | 8 | System display the options. | | | 9 | | Admin selects update option (teacher is assigned to which students). | | 10 | System display the data | | | 11 | | Admin edits and clicks update button. | | 12 | System verify the filled form and request for confirmation. | | | 13 | | Admin confirms. | | 14 | System updates the data and store in database. | | | Delete | | | | | | | | 17 | Admin clicks pat management button. | | 18 | | | System show the options. | | 19 | Admin selects delete option. | | 20 | | | System request for confirmation. | | 21 | Admin confirms to delete. | | 22 | | | System delete data from list and database. |   2 Alternative Flow   |  |  |  |  | | --- | --- | --- | --- | | S.N. | User action (if failed) | S.N. | System response | |  |  | 4.1 | If invalid teacher id is given system displays an error message and return to step 3 | |  |  | 2.1 and  4.1 | If system does not create and updates exam schedule, check if network is available or not. After connection, try again. | | |
| Post Conditions: User can view student information. | |
| Primary Actor: Admin | |
| Author: Everest | |

|  |  |
| --- | --- |
| Use Case Title | Announcement |
| Use Case ID | 7 |
| Description: To use system as an tool to announce information | |
| Pre-condition: Admin should be logged in | |
| Task sequence ­­  1 Normal flow:   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | S.N. | | User action | | S.N. | System response | | | Create | | | | | | | | 1 | | Admin clicks announcement button. | | 2 | System display the options. | | | 3 | | Admin select create option. | | 4 | System display form. | | | 5 | | Admin fills the form and click create button. | | 6 | System verify the form and store the data in database. | | | Update/ view | | | | | | | | 7 | | Admin clicks announcement button. | | 8 | System display the options. | | | 9 | | Admin selects update option. | | 10 | System display the data | | | 11 | | Admin edits and clicks update button. | | 12 | System verify the filled form and request for confirmation. | | | 13 | | Admin confirms. | | 14 | System updates the data and store in database. | | | Delete | | | | | | | | 17 | Admin clicks announcement management button. | | 18 | | | System show the options. | | 19 | Admin selects delete option. | | 20 | | | System request for confirmation. | | 21 | Admin confirms to delete. | | 22 | | | System delete data from list and database. |   2 Alternative Flow   |  |  |  |  | | --- | --- | --- | --- | | S.N. | User action (if failed) | S.N. | System response | |  |  | 4.1 | If invalid teacher id is given system displays an error message and return to step 3 | |  |  | 2.1 and  4.1 | If system does not create and updates exam schedule, check if network is available or not. After connection, try again. | | |
| Post Conditions: User can view student information. | |
| Primary Actor: Admin | |
| Author: Everest | |

|  |  |
| --- | --- |
| Use Case Title | Course |
| Use Case ID | 7 |
| Description: To display course material | |
| Pre-condition: Admin should be logged in | |
| Task sequence ­­  1 Normal flow:   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | S.N. | | User action | | S.N. | System response | | | Create | | | | | | | | 1 | | Admin clicks course management button. | | 2 | System display the options. | | | 3 | | Admin select create option. | | 4 | System display form. | | | 5 | | Admin fills the form and click create button. | | 6 | System verify the form and store the data in database. | | | Update/ view | | | | | | | | 7 | | Admin clicks course management button. | | 8 | System display the options. | | | 9 | | Admin selects update option. | | 10 | System display the data | | | 11 | | Admin edits and clicks update button. | | 12 | System verify the filled form and request for confirmation. | | | 13 | | Admin confirms. | | 14 | System updates the data and store in database. | | | Delete | | | | | | | | 17 | Admin clicks course management button. | | 18 | | | System show the options. | | 19 | Admin selects delete option. | | 20 | | | System request for confirmation. | | 21 | Admin confirms to delete. | | 22 | | | System delete data from list and database. |   2 Alternative Flow   |  |  |  |  | | --- | --- | --- | --- | | S.N. | User action (if failed) | S.N. | System response | |  |  | 4.1 | If invalid teacher id is given system displays an error message and return to step 3 | |  |  | 2.1 and  4.1 | If system does not create and updates exam schedule, check if network is available or not. After connection, try again. | | |
| Post Conditions: User can view student information. | |
| Primary Actor: Admin | |
| Author: Everest | |

Teacher

|  |  |
| --- | --- |
| Use Case Title | Log in |
| Use Case ID | 2 |
| Description: To use system an authorized user should log in using their assigned username and password. | |
| Pre-condition: all the user should use appropriate username and password and have an account. | |
| Task sequence  1 Normal flow:   |  |  |  |  | | --- | --- | --- | --- | | S.N. | User action | S.N. | System response | | 1 | Users go to login page | 2 | System open the login page. | | 3 | User enters their respective emails and password. (For new users, they must use their assigned password and username.) | 5 | If username and password is correct, then system is redirected to home page/ dashboard. | | 4 | Users then enter on system. |  |  |   2 Alternative Flow   |  |  |  |  | | --- | --- | --- | --- | | S.N. | User action (if failed) | S.N. | System response | |  |  | 5.1 | If error occurs, system will display error message (username or password is invalid). And system returns the step to 3 where user must fill the login form again. | | 5.2 | Users try again to login |  |  | | 5.3 | User fills the form or chooses the forget password. | 5.4 | System redirects page to forget password | | 5.5 | User fills necessary information to change password. |  |  | | 5.6 | Users then enter/click on change password button. |  |  | |  |  | 5.7 | System check user information from database and update their password and returns to step 3 where user can use their changed password. If any error is faced system return to step 5.5. | | |
| Post Conditions: If the ID or password is correct the user must be logged in. | |
| Primary Actor: Admin, Teacher, and Student | |
| Author: Everest | |

|  |  |
| --- | --- |
| Use Case Title | Timetable |
| Use Case ID | 4 |
| Description: To create and view timetable | |
| Pre-condition: Admin should be logged in | |
| Task sequence  1 Normal flow:   |  |  |  |  | | --- | --- | --- | --- | | S.N. | User action | S.N. | System response | | 3 | Teacher clicks view button. | 4 | System displays timetable. |   2 Alternative Flow   |  |  |  |  | | --- | --- | --- | --- | | S.N. | User action (if failed) | S.N. | System response | |  |  | 2.1 and  6.1 | If system does not create and updates timetable, check if network is available or not. After connection, try again. | | |
| Post Conditions: User can view timetable. | |
| Primary Actor: Admin | |
| Author: Everest | |

|  |  |
| --- | --- |
| Use Case Title | Exam |
| Use Case ID | 5 |
| Description: To display and create exam schedule | |
| Pre-condition: Admin should be logged in | |
| Task sequence  1 Normal flow:   |  |  |  |  | | --- | --- | --- | --- | | S.N. | User action | S.N. | System response | | 3 | Users click view button. | 4 | System displays timetable. |   2 Alternative Flow   |  |  |  |  | | --- | --- | --- | --- | | S.N. | User action (if failed) | S.N. | System response | |  |  | 2.1 and  6.1 | If system does not create and updates exam schedule, check if network is available or not. After connection, try again. | | |
| Post Conditions: User can view exam schedule. | |
| Primary Actor: Admin | |
| Author: Everest | |

|  |  |
| --- | --- |
| Use Case Title | Grade |
| Use Case ID | 6 |
| Description: To display student total marks and grade | |
| Pre-condition: Admin should be logged in | |
| Task sequence  1 Normal flow:   |  |  |  |  | | --- | --- | --- | --- | | S.N. | User action | S.N. | System response | | 1 | Users should declare result through feedback and grades. | 2 | System should display all grades provided by user. | | 3 | Users click view button. | 4 | System displays grades. | | 5 | User can change result (if necessary). | 6 | System updates the result. |   2 Alternative Flow   |  |  |  |  | | --- | --- | --- | --- | | S.N. | User action (if failed) | S.N. | System response | |  |  | 2.1 and  6.1 | If system does not create and updates exam schedule, check if network is available or not. After connection, try again. | | |
| Post Conditions: User can view their grades. | |
| Primary Actor: Admin | |
| Author: Everest | |

|  |  |
| --- | --- |
| Use Case Title | Student |
| Use Case ID | 7 |
| Description: To display student details | |
| Pre-condition: Admin should be logged in | |
| Task sequence  1 Normal flow:   |  |  |  |  | | --- | --- | --- | --- | | S.N. | User action | S.N. | System response | | 1 | Users should click student management. | 2 | System should all student information. | | 3 | Users enter student id and click search button. | 4 | System verifies student id and displays student information. |   2 Alternative Flow   |  |  |  |  | | --- | --- | --- | --- | | S.N. | User action (if failed) | S.N. | System response | |  |  | 4.1 | If invalid student id is given system displays an error message and return to step 3 | |  |  | 2.1 and  4.1 | If system does not create and updates exam schedule, check if network is available or not. After connection, try again. | | |
| Post Conditions: User can view student information. | |
| Primary Actor: Admin | |
| Author: Everest | |

|  |  |
| --- | --- |
| Use Case Title | attendance |
| Use Case ID | 7 |
| Description: To display student details | |
| Pre-condition: Admin should be logged in | |
| Task sequence  1 Normal flow:   |  |  |  |  | | --- | --- | --- | --- | | S.N. | User action | S.N. | System response | | 1 | Teacher clicks on attendance button. | 2 | System display option. | | 3 | Teacher selects record option. | 4 | System display attendance list and form. | | 5 | Teacher file check box until finish all students of class and click save button. | 6 | system verify the data and system record the attendance |   2 Alternative Flow   |  |  |  |  | | --- | --- | --- | --- | | S.N. | User action (if failed) | S.N. | System response | |  |  | 4.1 | If invalid student id is given system displays an error message and return to step 3 | |  |  | 2.1 and  4.1 | If system does not create and updates exam schedule, check if network is available or not. After connection, try again. | | |
| Post Conditions: User can view student information. | |
| Primary Actor: Admin | |
| Author: Everest | |

|  |  |
| --- | --- |
| Use Case Title | announcement |
| Use Case ID | 7 |
| Description: To view announcement using system | |
| Pre-condition: Admin should be logged in | |
| Task sequence  1 Normal flow:   |  |  |  |  | | --- | --- | --- | --- | | S.N. | User action | S.N. | System response | | 1 | Users click announcement. | 2 | System shows all the announcement. | | 3 | User clicks new announcement when they must announce. | 4 | System requests for confirmation. | | 5 | Teacher clicks confirm. | 6 | System stores the announcement. |   2 Alternative Flow   |  |  |  |  | | --- | --- | --- | --- | | S.N. | User action (if failed) | S.N. | System response | |  |  | 4.1 | If invalid teacher id is given system displays an error message and return to step 3 | |  |  | 2.1 and  4.1 | If system does not create and updates exam schedule, check if network is available or not. After connection, try again. | | |
| Post Conditions: User can view student information. | |
| Primary Actor: Admin | |
| Author: Everest | |

|  |  |
| --- | --- |
| Use Case Title | course |
| Use Case ID | 7 |
| Description: To view course content | |
| Pre-condition: Admin should be logged in | |
| Task sequence  1 Normal flow:   |  |  |  |  | | --- | --- | --- | --- | | S.N. | User action | S.N. | System response | | 1 | Users click course content. | 2 | System shows all the content. System requests to select which term. | | 3 | User selects. | 4 | System displays the user selected. |   2 Alternative Flow   |  |  |  |  | | --- | --- | --- | --- | | S.N. | User action (if failed) | S.N. | System response | |  |  | 4.1 | If invalid teacher id is given system displays an error message and return to step 3 | |  |  | 2.1 and  4.1 | If system does not create and updates exam schedule, check if network is available or not. After connection, try again. | | |
| Post Conditions: User can view student information. | |
| Primary Actor: Admin | |
| Author: Everest | |

|  |  |
| --- | --- |
| Use Case Title | Logout |
| Use Case ID | 3 |
| Description: to logout from system. | |
| Pre-condition: Login to the system. | |
| Task sequence  1 Normal flow   |  |  |  |  | | --- | --- | --- | --- | | S.N. | User action | S.N. | System response | | 1 | Users click logout button | 2 | System closes the dashboard and open the login page. |   2 Alternative flow   |  |  |  |  | | --- | --- | --- | --- | | S.N. | User action | S.N. | System response | |  |  | 2.1 | Check the network if user is not logged out and user should logout again. | | |
| Post Conditions: User is logged out from the system. | |
| Primary Actor: Admin, Teacher, and Student | |
| Notes: User should not forget logging out from system after logging in. | |
| Author: Everest | |

Student

|  |  |
| --- | --- |
| Use Case Title | Log in |
| Use Case ID | 2 |
| Description: To use system an authorized user should log in using their assigned username and password. | |
| Pre-condition: all the user should use appropriate username and password and have an account. | |
| Task sequence  1 Normal flow:   |  |  |  |  | | --- | --- | --- | --- | | S.N. | User action | S.N. | System response | | 1 | Users go to login page | 2 | System open the login page. | | 3 | User enters their respective emails and password. (For new users, they must use their assigned password and username.) | 5 | If username and password is correct, then system is redirected to home page/ dashboard. | | 4 | Users then enter on system. |  |  |   2 Alternative Flow   |  |  |  |  | | --- | --- | --- | --- | | S.N. | User action (if failed) | S.N. | System response | |  |  | 5.1 | If error occurs, system will display error message (username or password is invalid). And system returns the step to 3 where user must fill the login form again. | | 5.2 | Users try again to login |  |  | | 5.3 | User fills the form or chooses the forget password. | 5.4 | System redirects page to forget password | | 5.5 | User fills necessary information to change password. |  |  | | 5.6 | Users then enter/click on change password button. |  |  | |  |  | 5.7 | System check user information from database and update their password and returns to step 3 where user can use their changed password. If any error is faced system return to step 5.5. | | |
| Post Conditions: If the ID or password is correct the user must be logged in. | |
| Primary Actor: Admin, Teacher, and Student | |
| Author: Everest | |

|  |  |
| --- | --- |
| Use Case Title | Teacher |
| Use Case ID | 7 |
| Description: To display teacher details | |
| Pre-condition: Admin should be logged in | |
| Task sequence  1 Normal flow:   |  |  |  |  | | --- | --- | --- | --- | | S.N. | User action | S.N. | System response | | 1 | Users should click student management. | 2 | System should all teacher information. | | 3 | Users enter teacher id and click search button. | 4 | System verifies teacher id and displays teacher information. | | 6 | User clicks new/ delete teacher id. | 7 | User can create new teacher id. |   2 Alternative Flow   |  |  |  |  | | --- | --- | --- | --- | | S.N. | User action (if failed) | S.N. | System response | |  |  | 4.1 | If invalid teacher id is given system displays an error message and return to step 3 | |  |  | 2.1 and  4.1 | If system does not create and updates exam schedule, check if network is available or not. After connection, try again. | | |
| Post Conditions: User can view student information. | |
| Primary Actor: Admin | |
| Author: Everest | |

|  |  |
| --- | --- |
| Use Case Title | course |
| Use Case ID | 7 |
| Description: To view course content | |
| Pre-condition: Admin should be logged in | |
| Task sequence  1 Normal flow:   |  |  |  |  | | --- | --- | --- | --- | | S.N. | User action | S.N. | System response | | 1 | Users click course content. | 2 | System shows all the content. System requests to select which term. | | 3 | User selects. | 4 | System displays the user selected. |   2 Alternative Flow   |  |  |  |  | | --- | --- | --- | --- | | S.N. | User action (if failed) | S.N. | System response | |  |  | 4.1 | If invalid teacher id is given system displays an error message and return to step 3 | |  |  | 2.1 and  4.1 | If system does not create and updates exam schedule, check if network is available or not. After connection, try again. | | |
| Post Conditions: User can view student information. | |
| Primary Actor: Admin | |
| Author: Everest | |

|  |  |
| --- | --- |
| Use Case Title | Timetable |
| Use Case ID | 4 |
| Description: To create and view timetable | |
| Pre-condition: Admin should be logged in | |
| Task sequence  1 Normal flow:   |  |  |  |  | | --- | --- | --- | --- | | S.N. | User action | S.N. | System response | | 3 | Teacher clicks view button. | 4 | System displays timetable. |   2 Alternative Flow   |  |  |  |  | | --- | --- | --- | --- | | S.N. | User action (if failed) | S.N. | System response | |  |  | 2.1 and  6.1 | If system does not create and updates timetable, check if network is available or not. After connection, try again. | | |
| Post Conditions: User can view timetable. | |
| Primary Actor: Admin | |
| Author: Everest | |

|  |  |
| --- | --- |
| Use Case Title | Exam |
| Use Case ID | 5 |
| Description: To display and create exam schedule | |
| Pre-condition: Admin should be logged in | |
| Task sequence  1 Normal flow:   |  |  |  |  | | --- | --- | --- | --- | | S.N. | User action | S.N. | System response | | 3 | Users click view button. | 4 | System displays timetable. |   2 Alternative Flow   |  |  |  |  | | --- | --- | --- | --- | | S.N. | User action (if failed) | S.N. | System response | |  |  | 2.1 and  6.1 | If system does not create and updates exam schedule, check if network is available or not. After connection, try again. | | |
| Post Conditions: User can view exam schedule. | |
| Primary Actor: Admin | |
| Author: Everest | |

|  |  |
| --- | --- |
| Use Case Title | Grade |
| Use Case ID | 6 |
| Description: To display student total marks and grade | |
| Pre-condition: Admin should be logged in | |
| Task sequence  1 Normal flow:   |  |  |  |  | | --- | --- | --- | --- | | S.N. | User action | S.N. | System response | | 1 | Users should go to result through feedback and grades. | 2 | System request to choose which semester. | | 3 | Users click view button. | 4 | System displays grades. |   2 Alternative Flow   |  |  |  |  | | --- | --- | --- | --- | | S.N. | User action (if failed) | S.N. | System response | |  |  | 2.1 and  6.1 | If system does not create and updates exam schedule, check if network is available or not. After connection, try again. | | |
| Post Conditions: User can view their grades. | |
| Primary Actor: Admin | |
| Author: Everest | |

|  |  |
| --- | --- |
| Use Case Title | pat |
| Use Case ID | 6 |
| Description: To display student total marks and grade | |
| Pre-condition: Admin should be logged in | |
| Task sequence  1 Normal flow:   |  |  |  |  | | --- | --- | --- | --- | | S.N. | User action | S.N. | System response | | 1 | Users go to pat. | 2 | System should display all pat. | | 3 | Users click view button. | 4 | System displays assigned pat. | | 5 | User chooses when to meet their respective pat. | 6 | System saves the timetable. |   2 Alternative Flow   |  |  |  |  | | --- | --- | --- | --- | | S.N. | User action (if failed) | S.N. | System response | |  |  | 2.1 and  6.1 | If system does not create and updates exam schedule, check if network is available or not. After connection, try again. | | |
| Post Conditions: User can view their grades. | |
| Primary Actor: Admin | |
| Author: Everest | |

|  |  |
| --- | --- |
| Use Case Title | Logout |
| Use Case ID | 3 |
| Description: to logout from system. | |
| Pre-condition: Login to the system. | |
| Task sequence  1 Normal flow   |  |  |  |  | | --- | --- | --- | --- | | S.N. | User action | S.N. | System response | | 1 | Users click logout button | 2 | System closes the dashboard and open the login page. |   2 Alternative flow   |  |  |  |  | | --- | --- | --- | --- | | S.N. | User action | S.N. | System response | |  |  | 2.1 | Check the network if user is not logged out and user should logout again. | | |
| Post Conditions: User is logged out from the system. | |
| Primary Actor: Admin, Teacher, and Student | |
| Notes: User should not forget logging out from system after logging in. | |
| Author: Everest | |

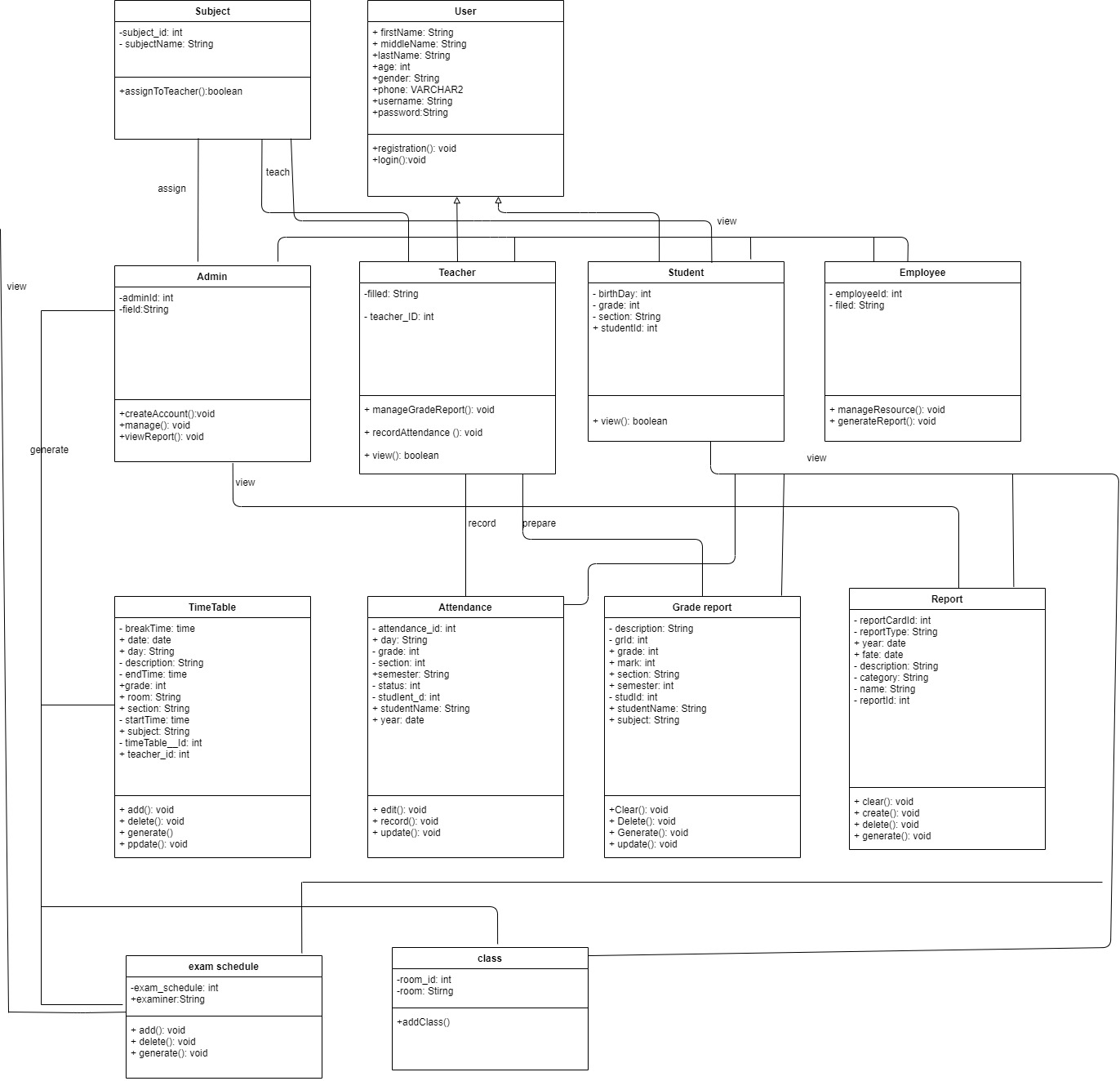


Fig: Class Diagram

Above is the class diagram of the system. A class diagram is a type of static structure Language which describes the structures of a system by showing the system's classes, attributes, operations and relation among the objects. Relations used in the diagrams include both composition and aggregation relationships.

**3.3** **Detailed dynamic system designs**

oThis portion contains the dynamic view of the system. In this Part it represents the system flow in advanced and proper way showing the use of methods,

**3.3.1.** **System sequence diagram**

It is an interaction diagram. These diagrams show how the operations are carried out in the system and when the messages are sent. These diagrams are managed according to the time.The objects involved in the operation of system are listed from left to right according to when they take part in the message sequence.

Sequence diagram is emphasizing the time ordering of the message and emphasize the structural organization of the object. They are helpful to identify the missing objects that are not identified in the analysis object mode. This allows the specification of is allows the specification of simple runtime scenarios in a graphical manner.

Below are some sequence diagrams for university website.

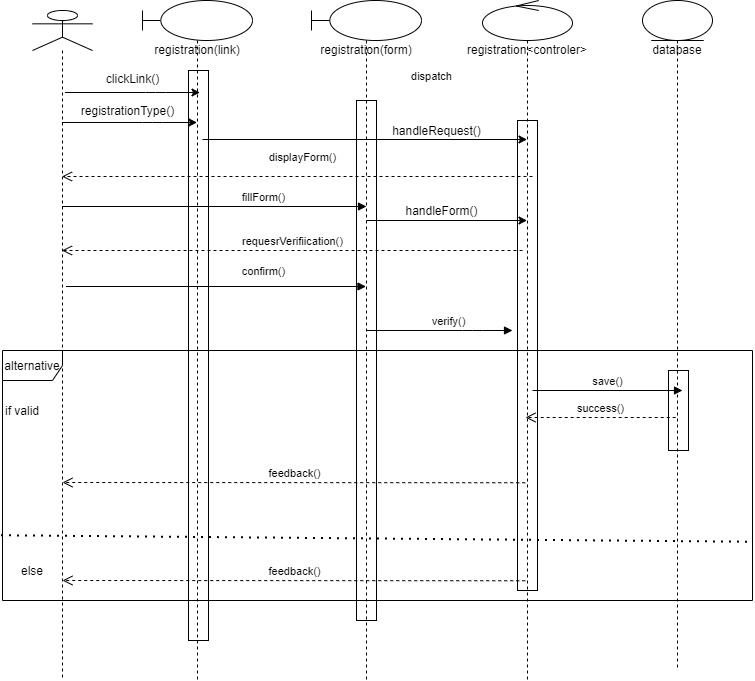
.

Fig: Sequence diagram for admin

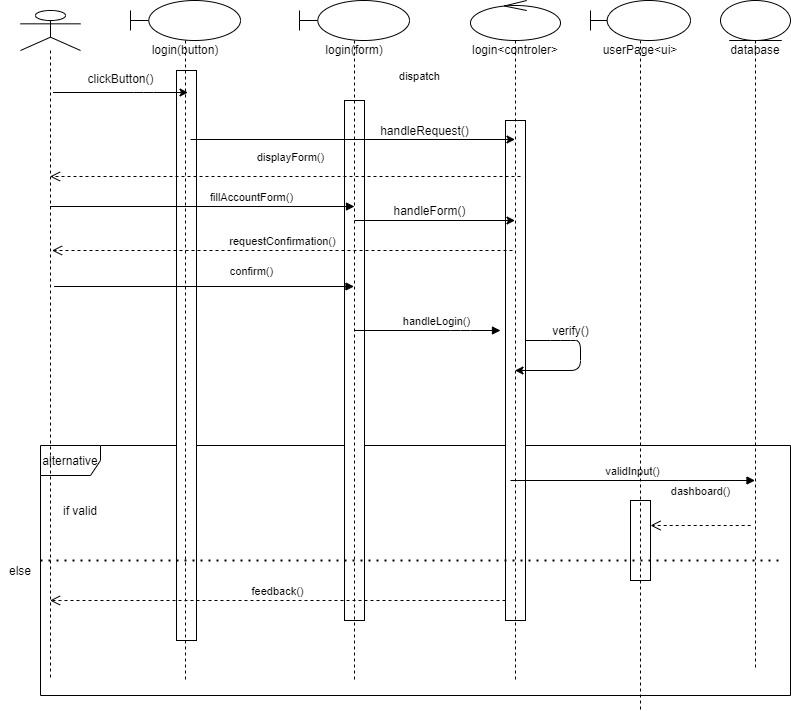


Fig: Sequence diagram for registration

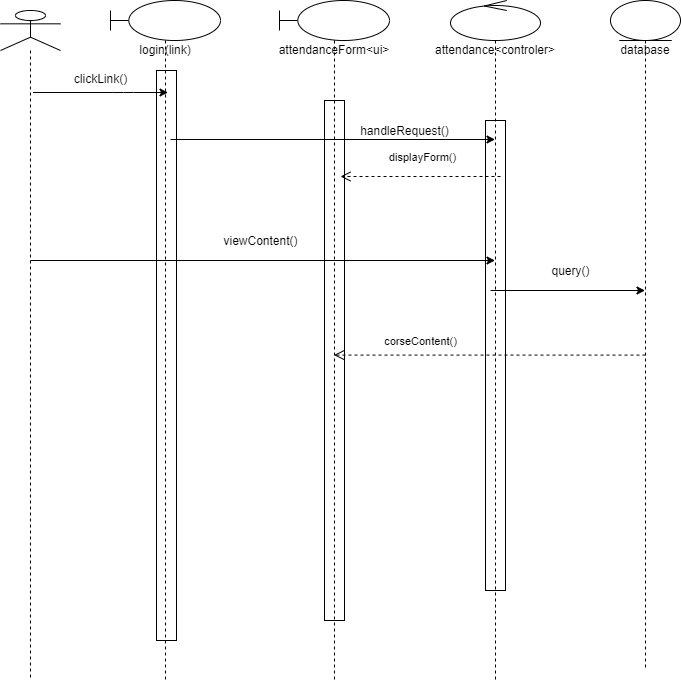


Fig: sequence diagram for course

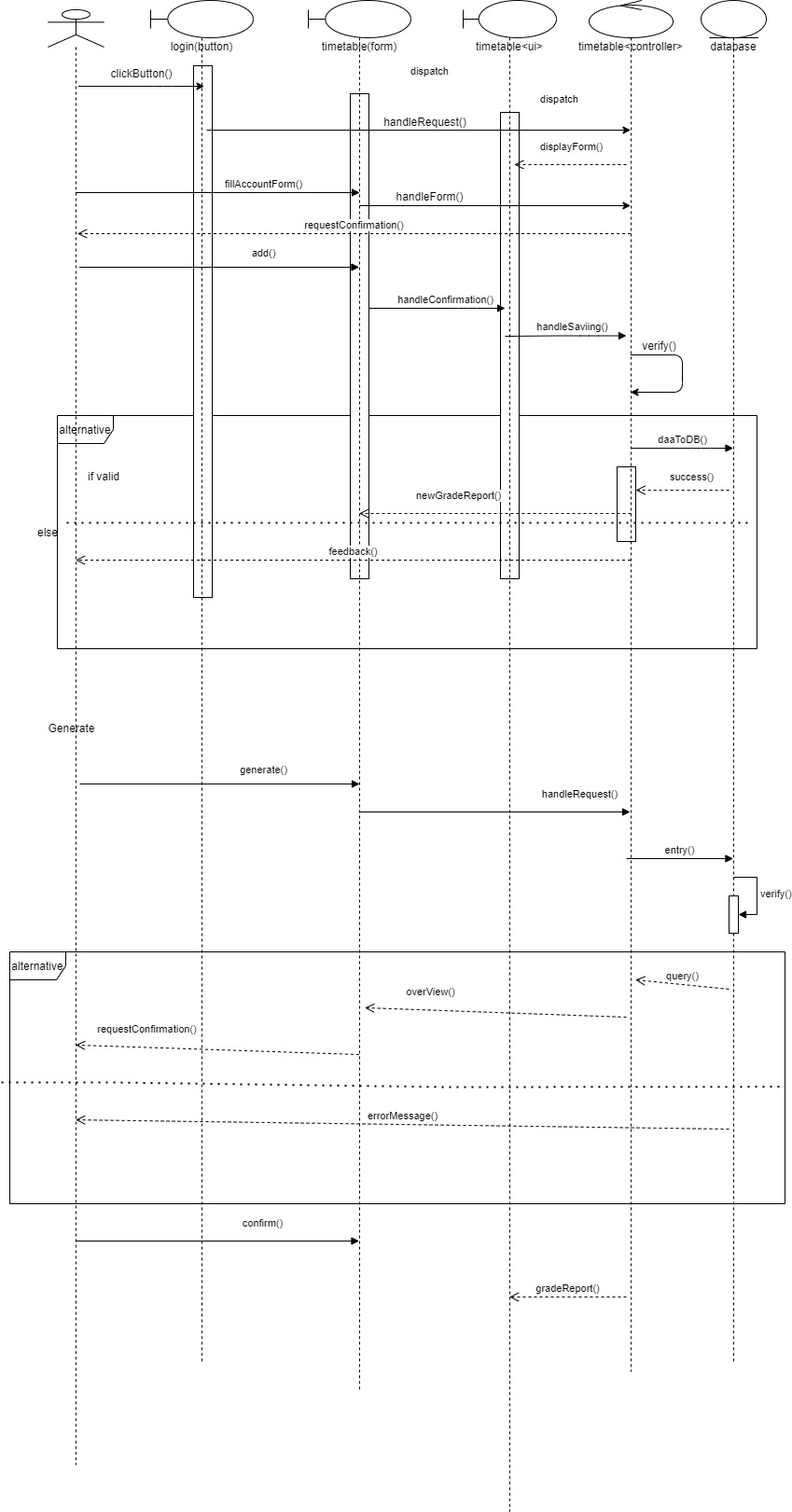


Fig: Sequence diagram for grade

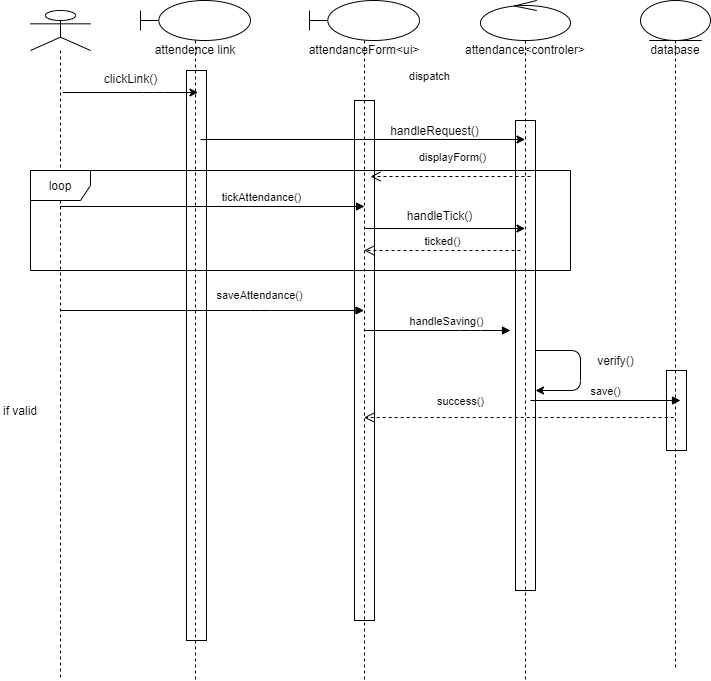


Fig: Sequence diagram for attendance

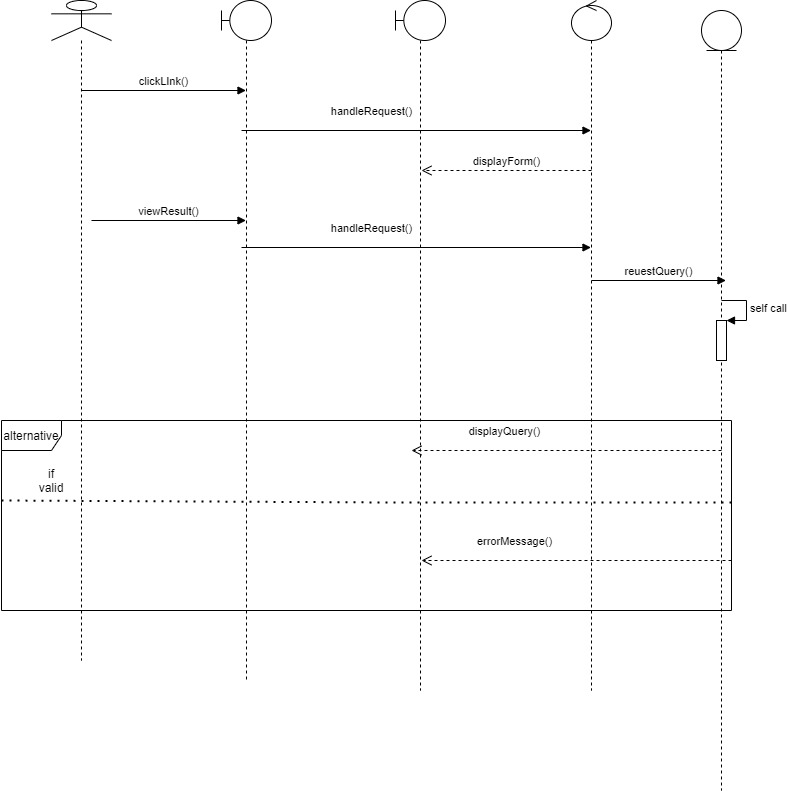


Fig: Sequence diagram for user report

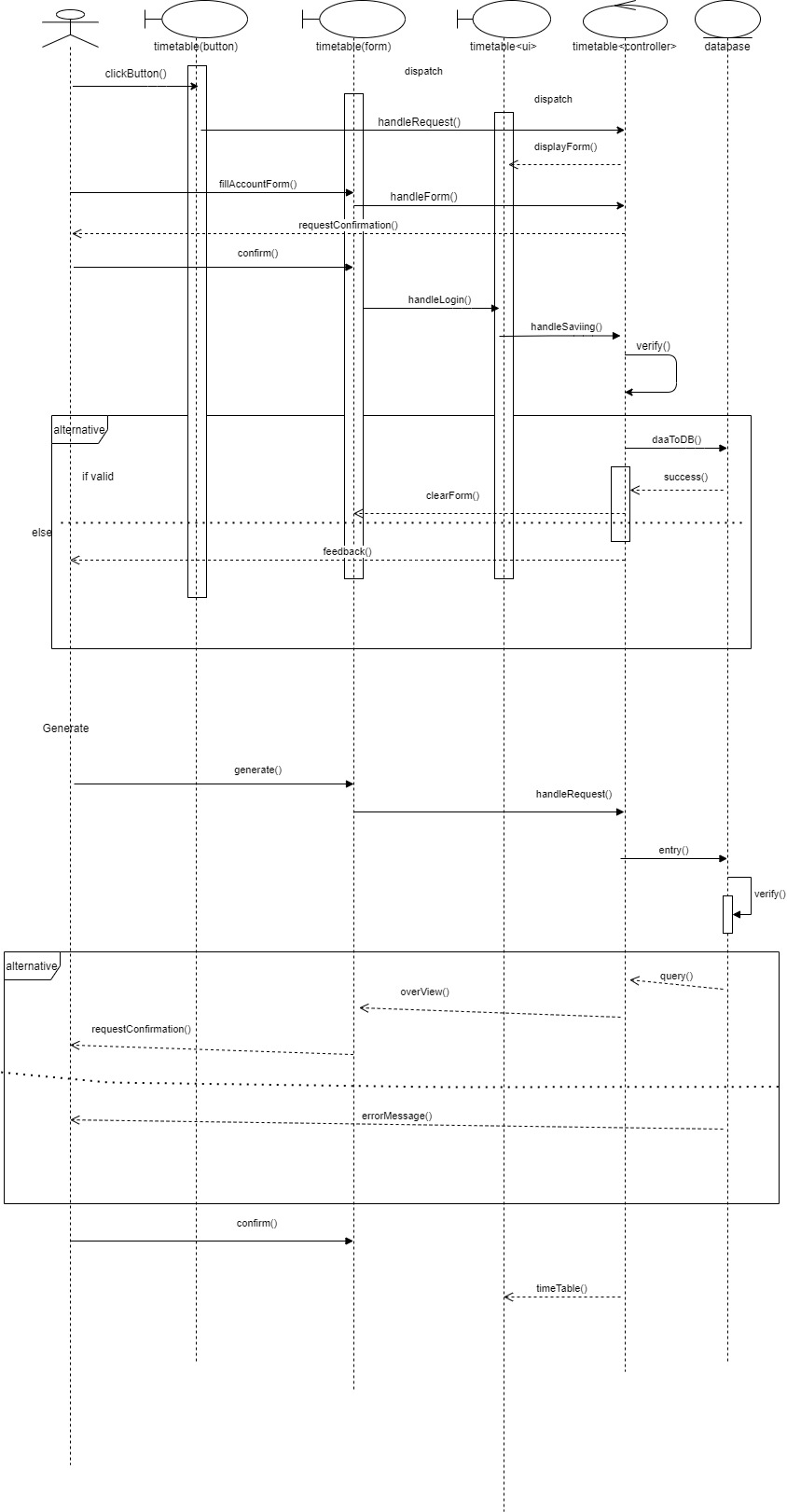


Fig: sequence diagram for timetable

DFD

It stands for data flow diagram which represents diagram of the data of the object in the system. It is also an way to show how the data flows or processed in the system in different stages.

. Data Flow Diagram serves two purposes:

1. To provide annunciation of how data are transformed as they move through the system.

2. To depict the functions that transforms the data flow.

Diagram

Description automatically generated

Diagram

Description automatically generated

**3.4.2. Attribute listing**

The below table shows all the attributes for the tables in the database

|  |  |  |
| --- | --- | --- |
| **S. no.** | **Table Name** | **Attributes** |
| 1. | Users | User id, Ufisrtname, UMiddlename, Ulastname, profile, contact, email, password, gender, bloodgroup, address, DOB, date, documents |
| 2. | Module-activity | Module-id, tutor-id, lecture slides, note, room-no, title |
| 3. | Module | m-id, Tutor-id, room-no, user-id, module-name |
| 4. | Courses | Std-id, course-id, user-id, pat-id, documents |
| 5. | Std-documents | Std-id, transcript, migration, mark-sheet, verification latter |
| 6. | Module-assignment | Module-id, assg\_title, create\_date, due­\_date, assg-brief, other-sources, |
| 7. | Time-table | Date, start-time, end-time, module-name, tutor-name, room-no,  Class-type |
| 8. | Attendance | Std-id, module-id, date, tutor-id, status |
| 9. | Grades | Std-id, exam-id, submitted-date, mark-scored, tutor-comment |
| 10. | Module activity | m-id, title, comments, activity |
| 11. | Announcement | User-id, ann-title, description, posted-date, display, |

**4.** **System Navigation Diagram**

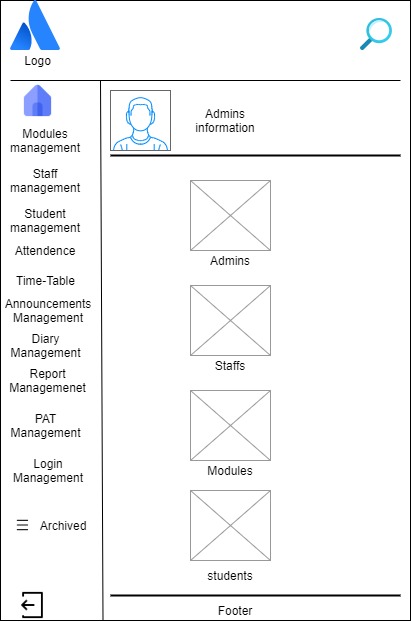
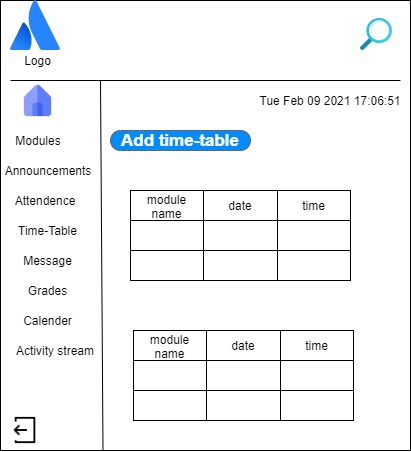
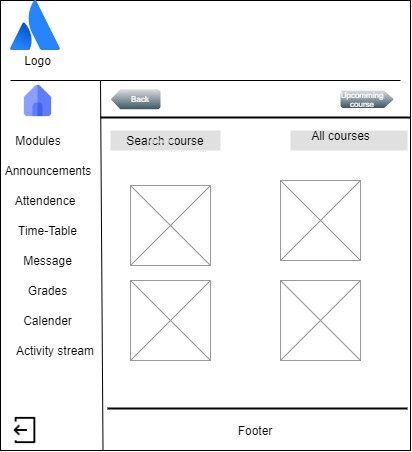
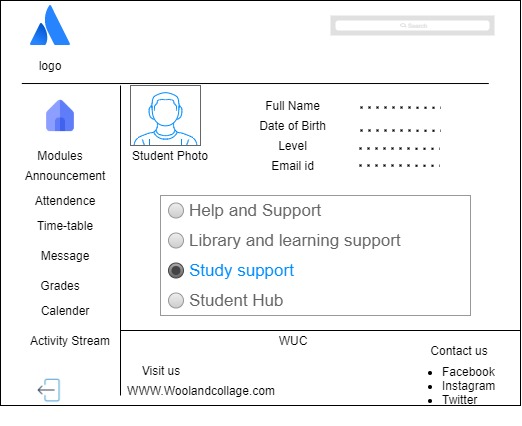
This is the visualization diagram for the system. In this process we use different methodologies in order to make implementation process easier and better to work. Some of the process used in this system are draft interface design, design revisions and heuristic evolution.

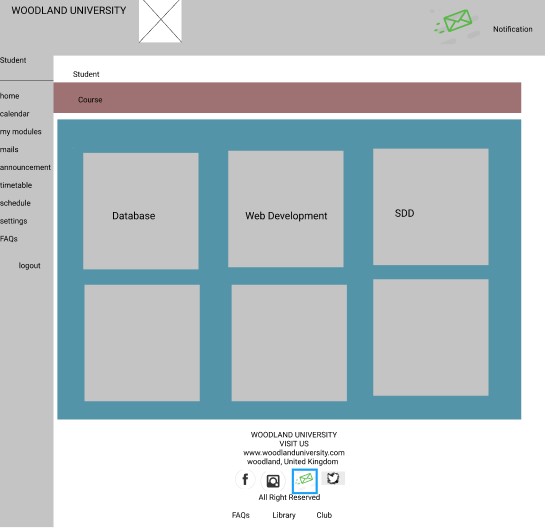
**4.1.** **Draft interface designs**

This process is done by making or designing wireframes, system mock-ups, activity event diagrams, navigation diagrams.

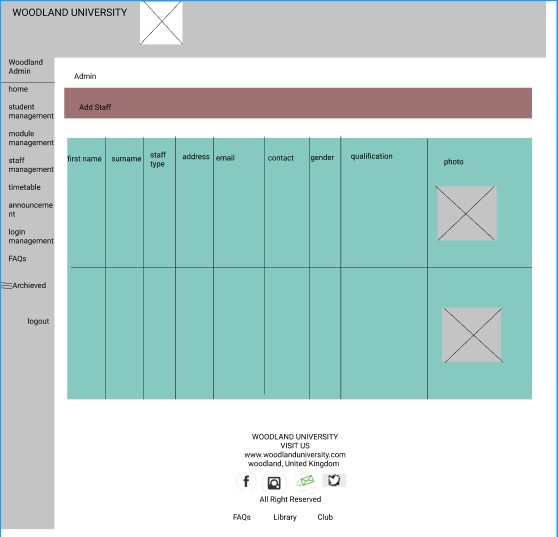
**4.1.1.** **Wireframes**

To build a website initially wireframes are the most important part. Wireframes are the simple sketch which provides the basic idea about the view of the system. It shows how the pages will look like and what content will it include. Wireframes contains the details which are important to a client.

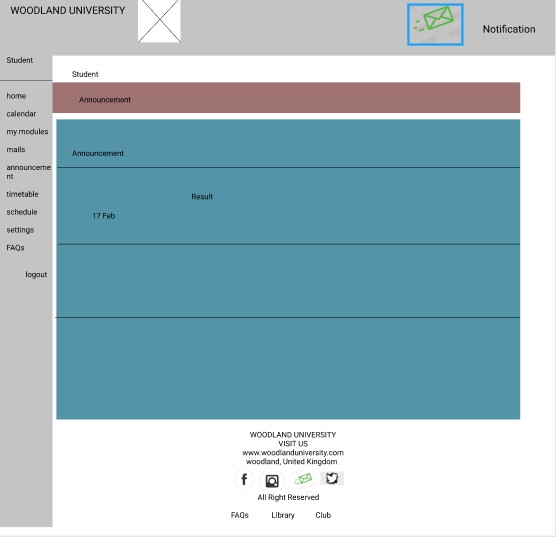


**

*Fig: wireframe for course content page*

**

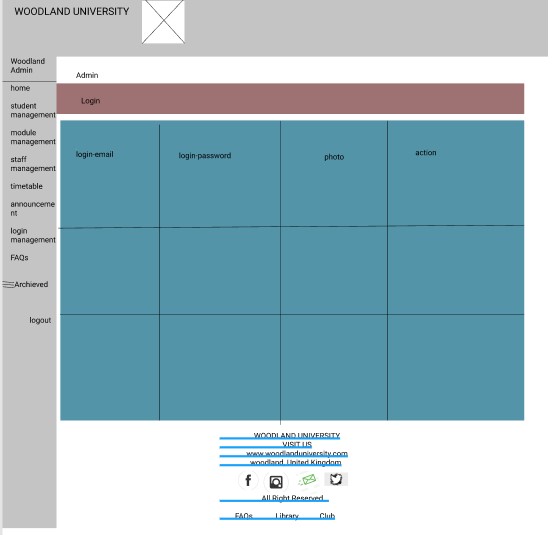
*Fig: to add staffs*



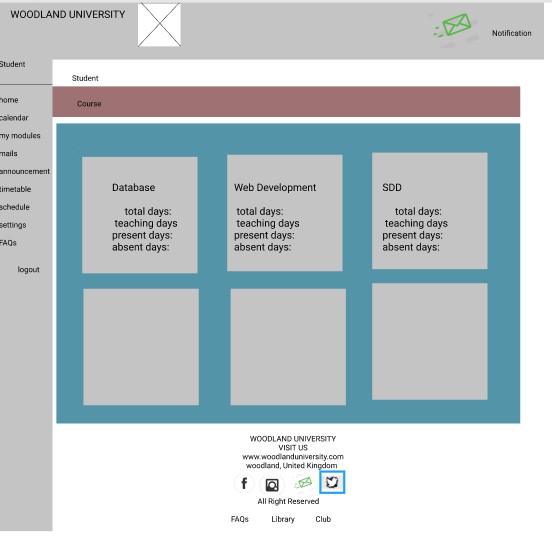
*Fig: to add announcement*

**

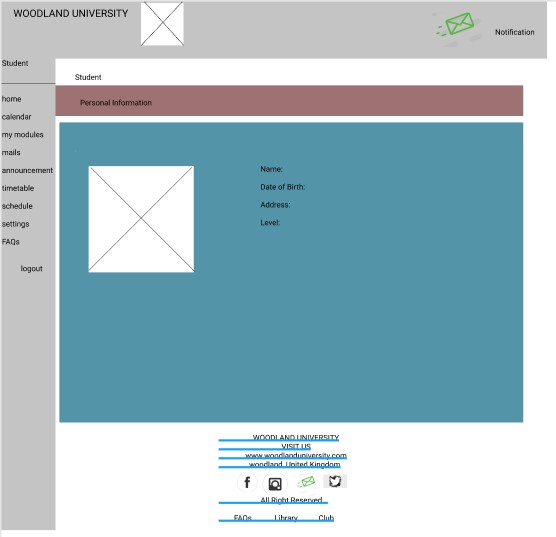
*Fig: to write mails*

**

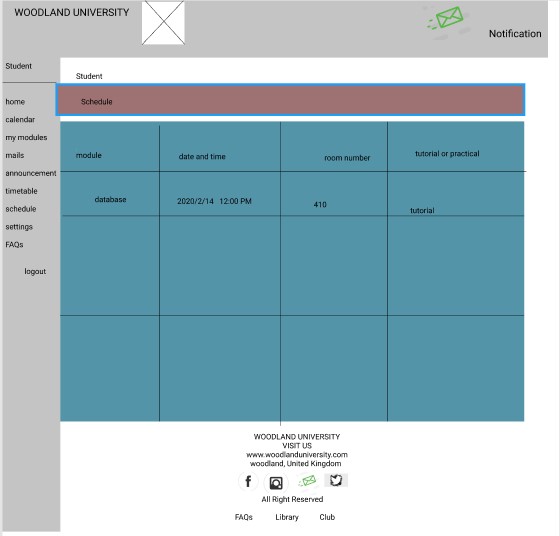
*Fig: login edit page for admin*

**

*Fig: attendance page*

**

*Fig: student page*

**

*Fig: time table page*

**4.1.2 System Navigation Page**

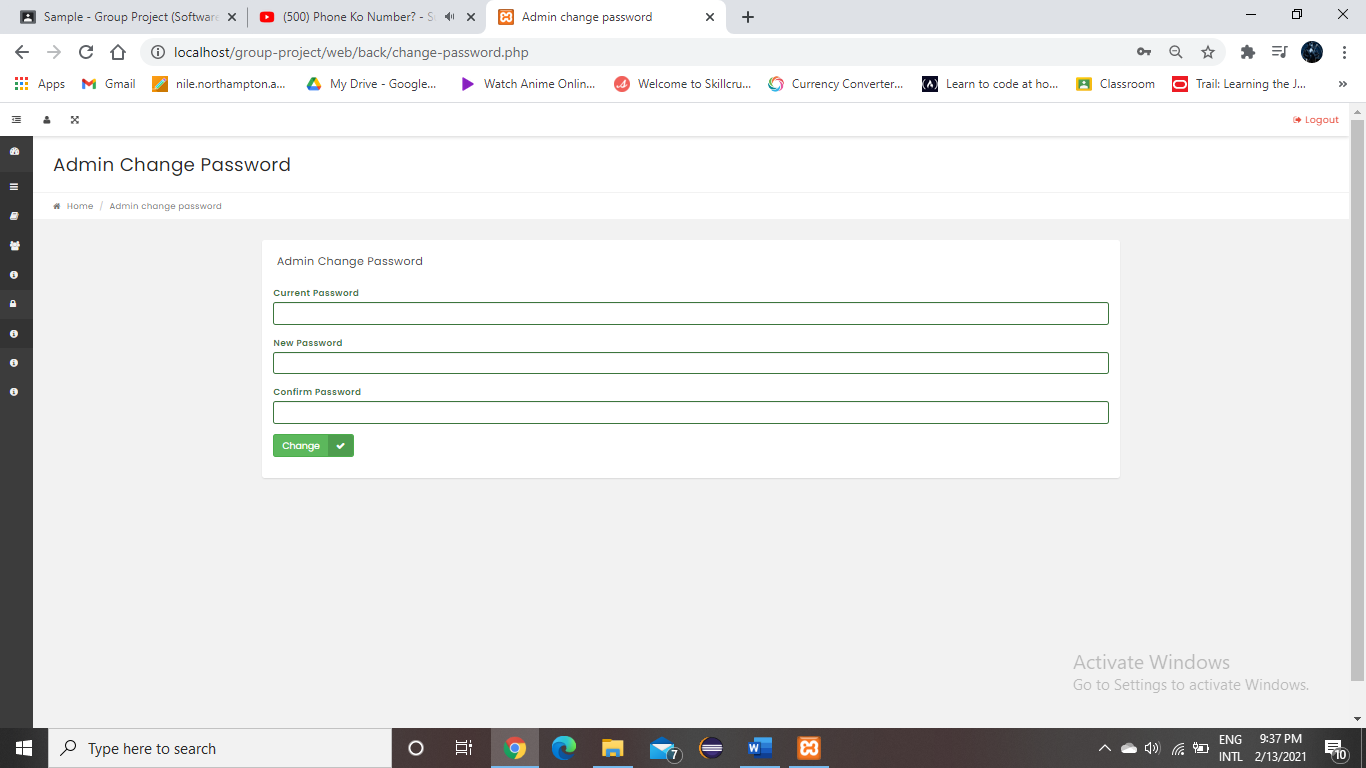
System Navigation Diagrams shows the stream of system. The main objective of these diagrams is to show the links that connects the system and helps to work the system in proper way. It shows the pathway of working methodology.

4.1.3 system mock-ups

Mock-ups is the prototype which gives the preview of how the final design will look like. Mock-ups are used: to show the overall content of the system developed with the contents also with the font used and color used while developing the software.

Graphical user interface

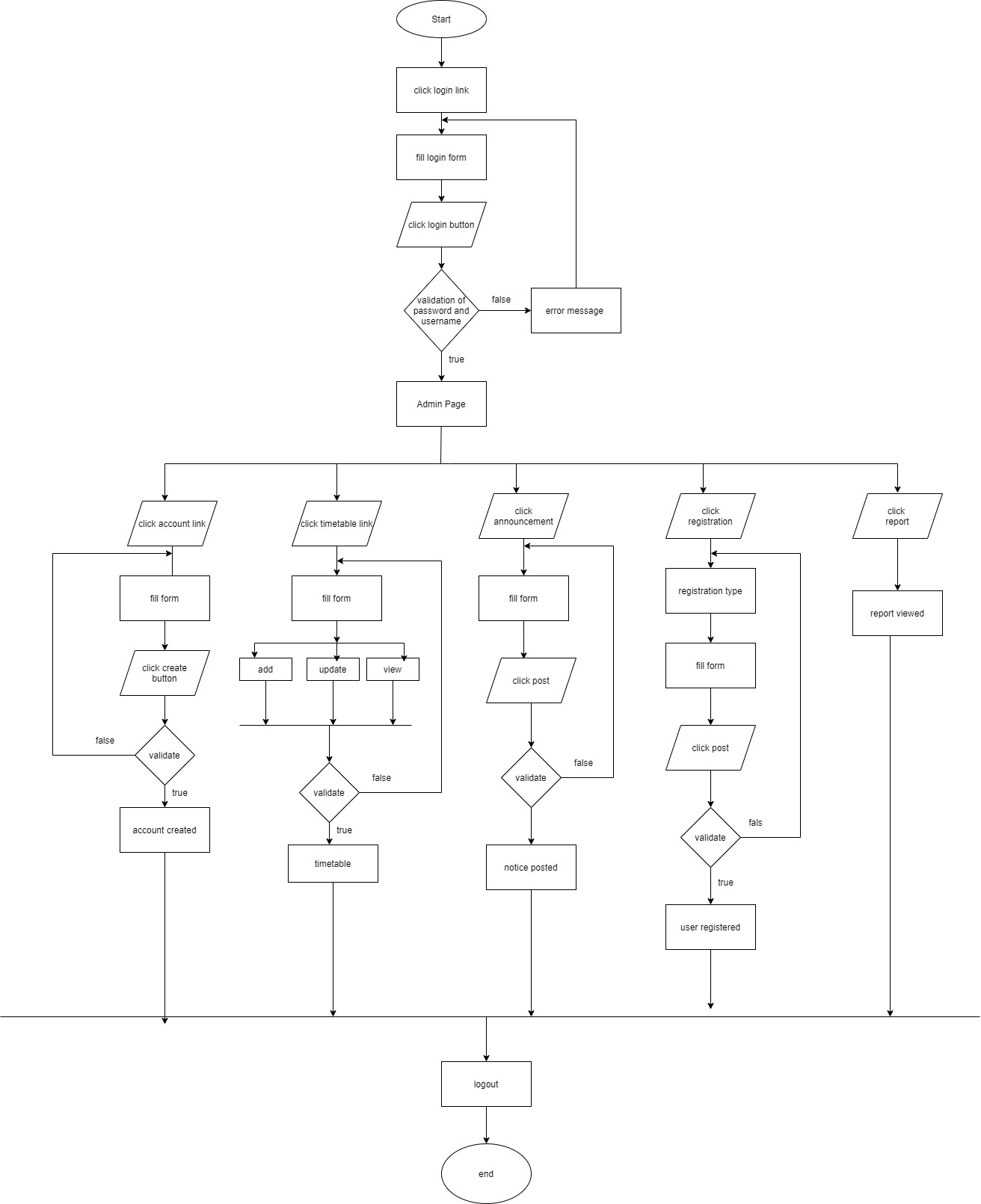
Description automatically generated fig the landing page for woodland university containing the image of college and courses module



**System Activity Event Diagrams**

Activity diagram is the graphical representation of workflows of graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the UML (Unified Modeling Language), activity diagrams can be used to describe the business describe the business and operational step-by-step workflows of components in a system. It shows the overall flow of control constructed from limited number of shapes connected with arrows.

It shows the flow of system in the form of the flow charts. The specific events of the system are represented by the flow charts.

**** Fig: flowchart showing the activity diagram

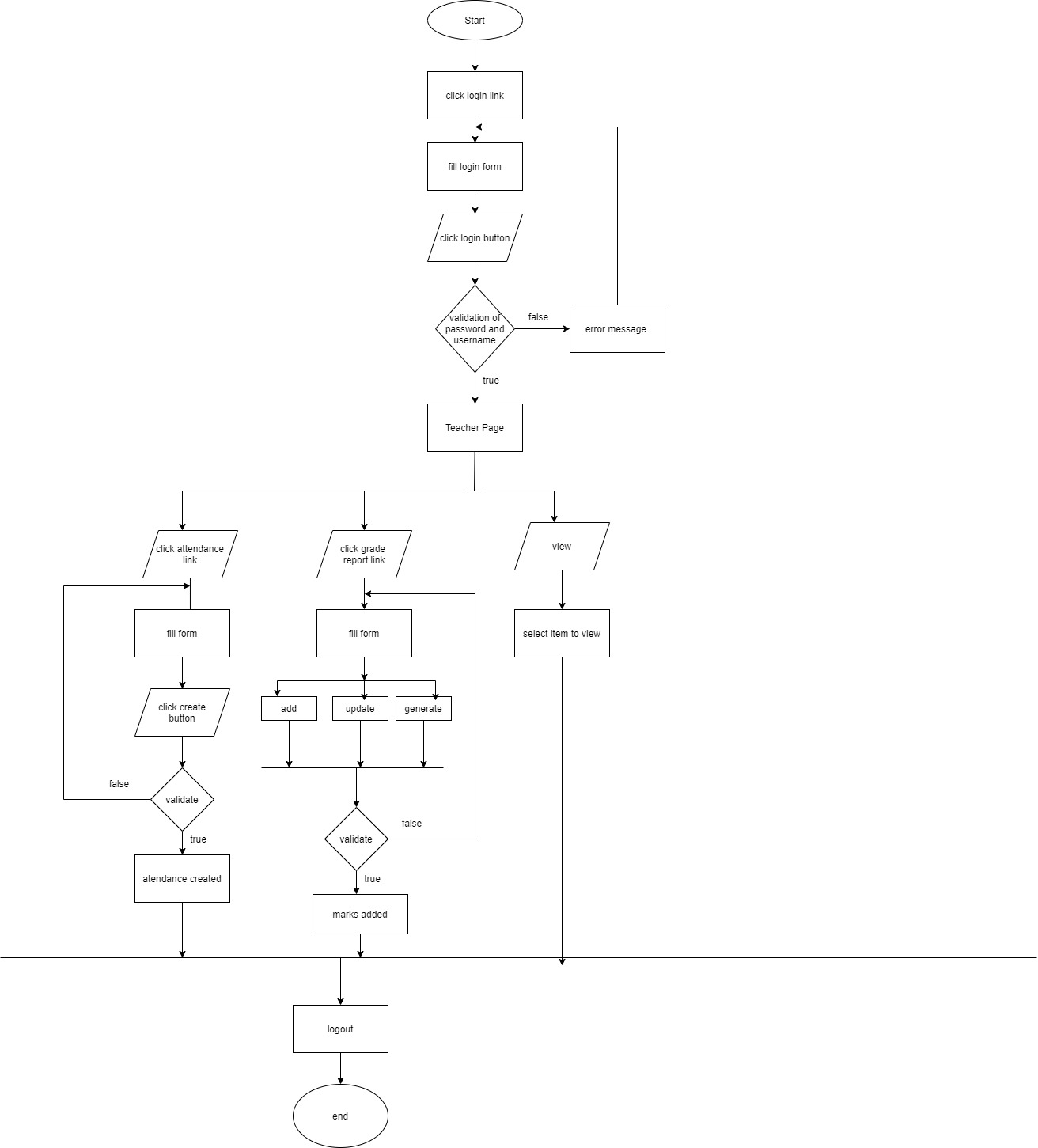


Fig: flowchart showing activity of teacher

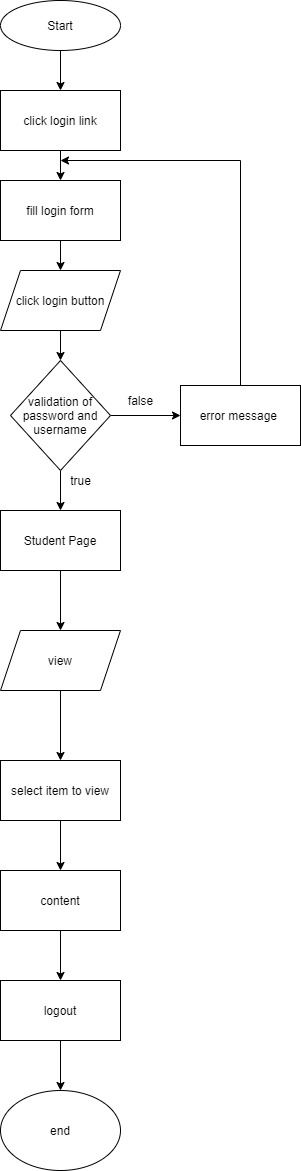


Fig: flowchart showing activity of students

Diagram

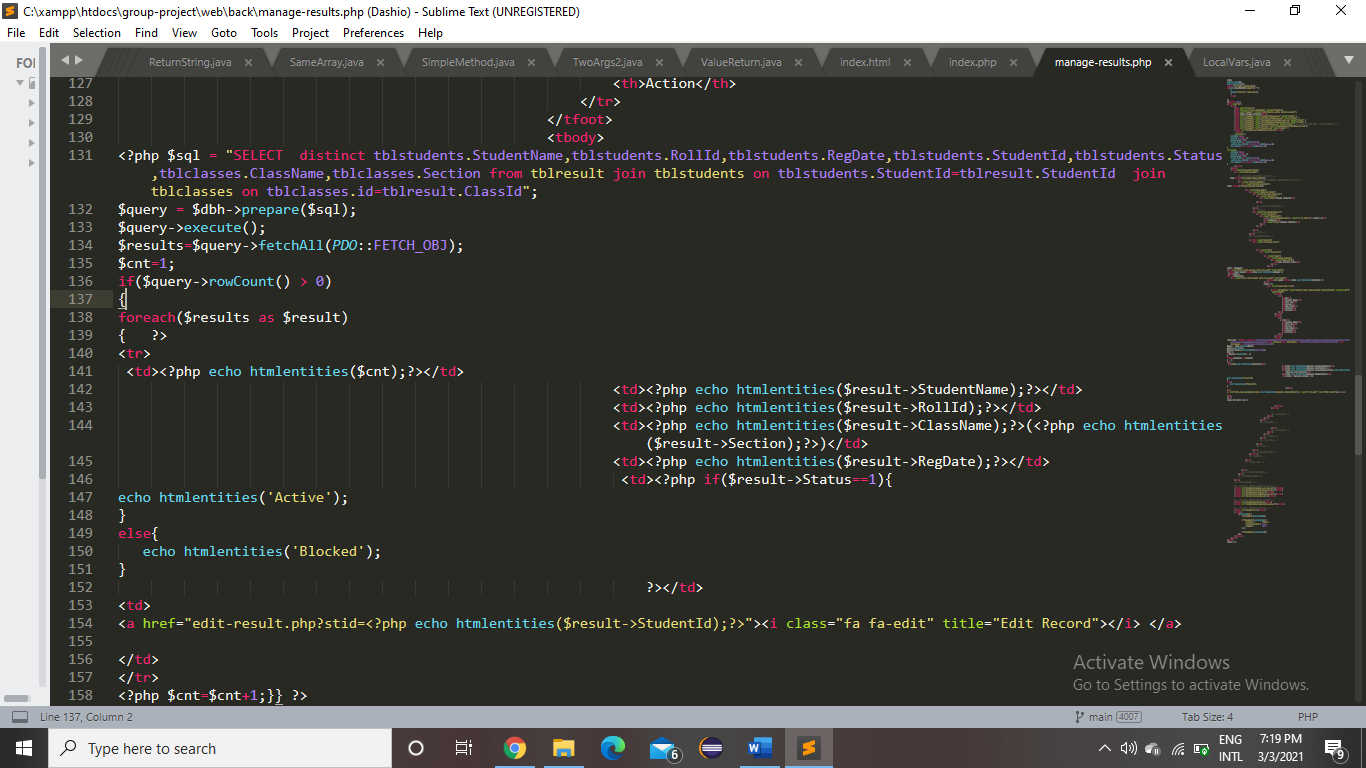
Description automatically generated

Chart

Description automatically generated

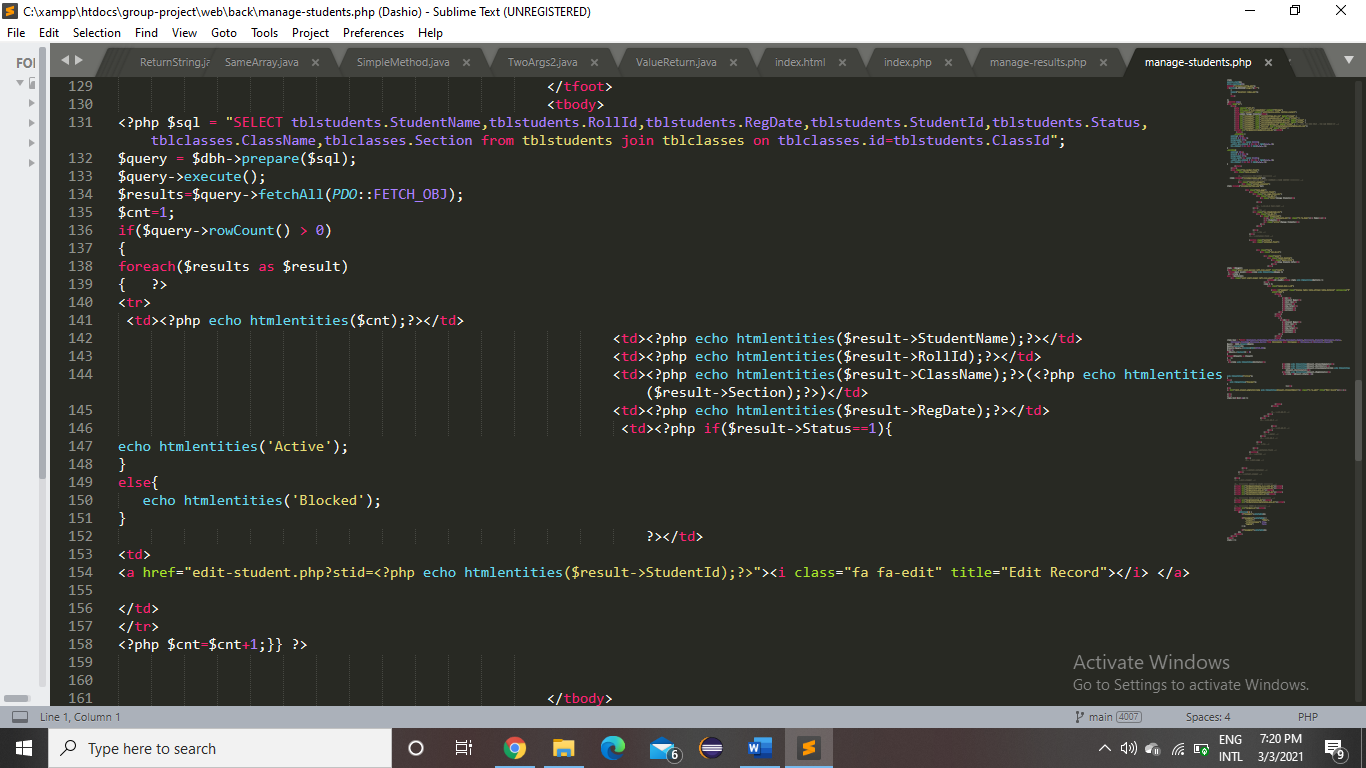
1. Technical difficult code sections

1 managing result

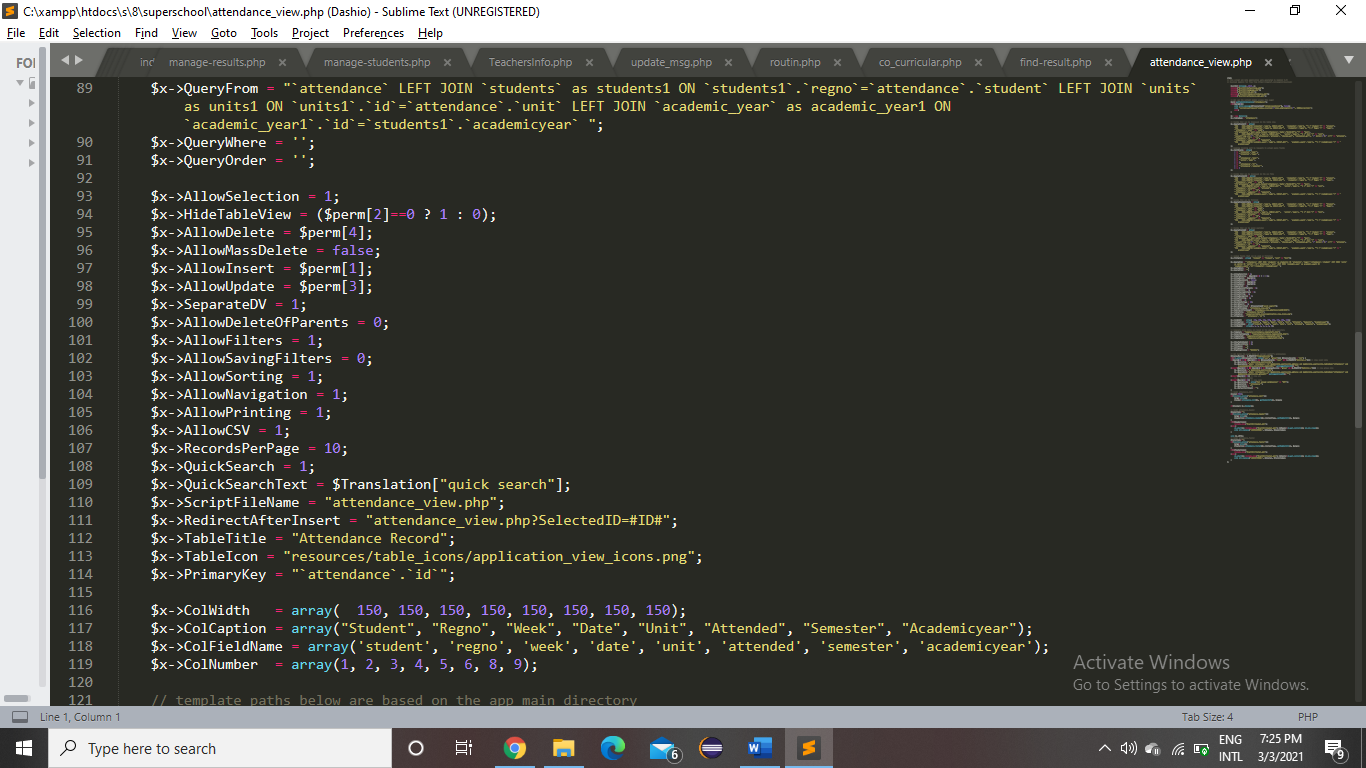


manage result

functionalities



Manage student



Attendance view

After the development of system, it need to be tested. As we know, no system is perfect so to make the system up to perfect testing is done. It is one of the most important part of system development life cycle. It is used so that the system works correctly and push the system to its maximum. Before user will be handed the system, all the error must be eliminated before the system, so, the testing is important to the success or completion of the system. Testing technique includes finding bugs or defect but are not only limited to it, it also includes the process of running a programs which provies an independent view to allow the business and the risks for implementaing the system.

The developed system is tested using unit testing (PHP Unit testing and black box testing) to find if the error faced are either one of syntax, logical or run time error.

Unit testing

It is a technique to test the every component of software(school management system) where it checks if there is any issue or bugs present in the system. It is used to check for the correctness of code and for testing it to test if every required function works or not. PHPUNIT is used in unit testing for this project which will be done for the system developed in PHP. It is done during the coding phase of an application. Doing so will help in costs reduction. Its success rate is great in detecting the issue in system. Using unit testing will allows automation of the testing and reduces difficulty in finding the errors in the system. The easiest way to use PHPUNIT is to download PHAR file and installing in system environment variable by giving the location of phar file which will be added in php of xampp folder. For testing of php file class name should have same name as file name and extend the file PHPUnit/FrameWork/Testcase.

Black Box Testing

It is a software testing technique in which functionality of the software under test (SUT) is tested and tester knows nothing about what is being tested or without looking at the internal code structure.

**7. System Evaluation**

**7.1. Usability Evaluation Strategy**

The process of judgment of system's usability and elasticity is called Evaluation. In the beginning a series of questions is set to take an interview with the client. In this session the client is asked about the functionality of the system, how he/she wants their system to be designed. After the system is designed the system is further asked about the system performance after he/she uses it. Some of the sample questions are discussed below.

1. On the whole, how was your experience using the system?
2. Did you face any complication?
3. Are all functionality working as per you need?
4. Did you found any errors in the system?
5. Do you need any help using the system?

**7.2. System Pilot Trials**

**User 1:**

1. How was your experience using the system?

|  |  |
| --- | --- |
| Excellent |  |
| Good | Green Check Mark Icon In A Circle. Tick Symbol In Green Color,.. Royalty  Free Cliparts, Vectors, And Stock Illustration. Image 136058493. |
| Satisfactory |  |
| Bad |  |

1. Are all functionality working as per you need?

|  |  |
| --- | --- |
| Yes | Green Check Mark Icon In A Circle. Tick Symbol In Green Color,.. Royalty  Free Cliparts, Vectors, And Stock Illustration. Image 136058493. |
| No |  |

1. Did you face any complication?

|  |  |
| --- | --- |
| Yes | Green Check Mark Icon In A Circle. Tick Symbol In Green Color,.. Royalty  Free Cliparts, Vectors, And Stock Illustration. Image 136058493. |
| No |  |

1. Did you found any errors in the system?

|  |  |
| --- | --- |
| Yes | Green Check Mark Icon In A Circle. Tick Symbol In Green Color,.. Royalty  Free Cliparts, Vectors, And Stock Illustration. Image 136058493. |
| No |  |

1. Do you need any help using the system?

|  |  |
| --- | --- |
| Yes | Green Check Mark Icon In A Circle. Tick Symbol In Green Color,.. Royalty  Free Cliparts, Vectors, And Stock Illustration. Image 136058493. |
| No |  |

**User 2:**

1. How was your experience using the system?

|  |  |
| --- | --- |
| Excellent | Green Check Mark Icon In A Circle. Tick Symbol In Green Color,.. Royalty  Free Cliparts, Vectors, And Stock Illustration. Image 136058493. |
| Good |  |
| Satisfactory |  |
| Bad |  |

1. Are all functionality working as per you need?

|  |  |
| --- | --- |
| Yes | Green Check Mark Icon In A Circle. Tick Symbol In Green Color,.. Royalty  Free Cliparts, Vectors, And Stock Illustration. Image 136058493. |
| No |  |

1. Did you face any complication?

|  |  |
| --- | --- |
| Yes |  |
| No | Green Check Mark Icon In A Circle. Tick Symbol In Green Color,.. Royalty  Free Cliparts, Vectors, And Stock Illustration. Image 136058493. |

1. Did you found any errors in the system?

|  |  |
| --- | --- |
| Yes | Green Check Mark Icon In A Circle. Tick Symbol In Green Color,.. Royalty  Free Cliparts, Vectors, And Stock Illustration. Image 136058493. |
| No |  |

1. Do you need any help using the system?

|  |  |
| --- | --- |
| Yes |  |
| No | Green Check Mark Icon In A Circle. Tick Symbol In Green Color,.. Royalty  Free Cliparts, Vectors, And Stock Illustration. Image 136058493. |

**User 3:**

1. How was your experience using the system?

|  |  |
| --- | --- |
| Excellent |  |
| Good |  |
| Satisfactory | Green Check Mark Icon In A Circle. Tick Symbol In Green Color,.. Royalty  Free Cliparts, Vectors, And Stock Illustration. Image 136058493. |
| Bad |  |

1. Are all functionality working as per you need?

|  |  |
| --- | --- |
| Yes | Green Check Mark Icon In A Circle. Tick Symbol In Green Color,.. Royalty  Free Cliparts, Vectors, And Stock Illustration. Image 136058493. |
| No |  |

1. Did you face any complication?

|  |  |
| --- | --- |
| Yes | Green Check Mark Icon In A Circle. Tick Symbol In Green Color,.. Royalty  Free Cliparts, Vectors, And Stock Illustration. Image 136058493. |
| No |  |

1. Did you found any errors in the system?

|  |  |
| --- | --- |
| Yes | Green Check Mark Icon In A Circle. Tick Symbol In Green Color,.. Royalty  Free Cliparts, Vectors, And Stock Illustration. Image 136058493. |
| No |  |

1. Do you need any help using the system?

|  |  |
| --- | --- |
| Yes | Green Check Mark Icon In A Circle. Tick Symbol In Green Color,.. Royalty  Free Cliparts, Vectors, And Stock Illustration. Image 136058493. |
| No |  |

**7.3. System Trial Results**

From the feedback result of the users we have made following conclusions:

|  |  |  |
| --- | --- | --- |
| S.no. | System Evaluation | Average Evaluation Results |
| 1. | System Usability | Simple To Use |
| 2. | System complication | Good |
| 3. | Errors | No errors |
| 4. | System Functionalities | Proper as expected |
| 5. | Help for system use | Required |
| 6. | Overall feedback | Good |

**8. Project Conclusions**

There may be possibility of some limitation in this project. It was developed for the Woodland University to manage the records, course content and to take the attendance to user which will boost the overall process in the university. From the elicitation activities we got the overview of the system and we work on the project and coinducteed different meeting with the client and with the agreement and the successful running of the code. This project contains the most of the area required for the system but few area are not done due to the lack of knowledge of the user and less time provided. But the area contain in the system works fine and will efficiently run in the university.

Nevertheless, the developed system will run smoothly with the project having the functionality of adding, updating/ amending , and deleting for the most part of the area. It also contain few area to be archived for the future use and also has the features of assigning and marking according to the elicitation activities. The project has been completed using PHP around for completion.

The main user of the system Mr. Adam Blake who acts as an administrator of the system will have the following section to look after with the functionality:

|  |  |
| --- | --- |
| Area | Notes/Essential Functionality |
| Student Records | Create, Amend, Archive, Display, Assign. |
| Staff Records | Create, Amend, Archive, Display, Assign. |
| Course Records | Create Structure, Amend, Display, Delete, Archive. |
| Module Management | Create, Amend, Delete, Archive, Display, Assign. |
| Timetable Management | Create, Amend, Delete, Archive, Display |
| Announcement | Create, amend, delete |

The second user is Dr. Raj Singh who is an module leader and will be looking after following area with functionality

|  |  |
| --- | --- |
| Area | Notes/Essential Functionality |
| Module | Create, Amend, Archive, Display, Assign. |
| Assign,ent | Create, Amend, Archive, Display, Assign. |
| Submissiona | Create Structure, Amend, Display, Delete, Archive. |
| resource | Create, Amend, Delete, Archive, Display, Assign. |
| attendance | Create, Amend, Delete, Archive, Display |
| Announcement | Create, amend, delete |
| pat |  |

Another user for the system is Mr. Mark Williams who is an student registered in Woodland University reading computing course module.

Module

Announcements

Grades

Forum

The developed project is an web based for managing the overall college management. This system has been completed successfully and was tested using blackbox and unit testing using php. The developed system is user friendly and has most area covered to perform the task done in course management system which is saved in database and can be retrieved by the users.

The given project is an small foray into the field of programing where we learnt about test case and how to use API. It also help us to learn to work the team and how to lead a team.

Project management

Project management is the method of driving the work of a group to attain objectives and meet good result criteria at a specified time. The essential challenge of project administration is to attain all the project objectives inside the given elicitation.

The project is managed using trello and github. All the group members have put their task in github. Different cards were made in trello where the assigned task for all the members as put and members have given their update. Similarly, in github the assigned code was pushed.

Due to the proper help of our respected teacher Mr. Suresh Gautam, we were able to finish this project within given time. While working on this project we were able to learn about API’s, test cases, BON chart and many other things related to the assignment. With the help of this project, we have learnt about group management and the working procedure with group within a industry.

Below are different attributes that define the proper process for the management of project.

Gantt chart

Gantt chart is an type of bar which gives an idea of time spent in specific area. It gives an overview how much time is spent to in though out the project. In vertical, it shows the tasks performed and in horizontal it shows the time interval.

Chart

Description automatically generated with medium confidence

A picture containing graphical user interface

Description automatically generated

**9.3 Project Quality Plan/Strategy**

A document which specify the quality, practices, standards, resources, and the sequence of activities relevant to a particular product, service, project, or contract is called the quality plan. The more the necessities of the project is fulfilled then it is measured as the more qualitative work. The main aim and objectives of the project plan is to complete the requirements presented by the client. All the functionalities could not be achieved while doing the project due to the limited time or lack of resources.

All the functionalities completed in the system and the plan are listed in the table below :

|  |  |  |  |
| --- | --- | --- | --- |
| S.N | Functionality | Importance Level | Completion Status |
| Administrator | | | |
| 1. | Adding administrators, module leaders into the system | Very High | Completed |
| 2. | Assigning the PAT to students | High | Completed |
| 3. | Assigning a module to a module leader | Very High | Completed |
| 4. | Adding announcements | Very High | Completed |
| 5. | Amending the status of the students | High | Completed |
| 6. | Assigning the courses to students | Very High | Completed |
| 7. | Adding new courses and modules | Moderate | Completed |
| 8. | Creating and Amending the courses | Very High | Completed |
| 9. | Creating and Amending the modules | Very High | Completed |
| 10. | Creating and Amending the students | Very High | Completed |
| Module Leader | | | |
| 11. | Adding the module activities | Very High | Completed |
| 12. | Assigning assignments to the students | High | Completed |
| 13. | Marking the grades of the assignment | Very High | Completed |
| 14. | Monitoring the attendance of the students | Very High | Completed |
| 15. | Adding announcements | High | Completed |
| 16. | Displaying the grades | Very High | Completed |
| 17. | Generating reports | High | Completed |
| Students | | | |
| 18. | Submitting assignment | Very High | Completed |
| 19. | Viewing grades | Very High | Completed |
| 20. | Sending email to PAT | Very High | Completed |
| 21. | Viewing the announcements | Very High | Completed |
| 22. | Viewing the timetable | Very High | Completed |

**9.4 Project Work Log**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Task | Individual Contribution | | | |
| **Introduction** | **Inshan K.C** | **Milan Khatiwada** | **Pankaj Badu** | **Ronak Bastola** |
| Project Background | 25% | 25% | 25% | 25% |
| Project Aims and Objectives | 25% | 25% | 25% | 25% |
| Project Development Methodology | 25% | 25% | 25% | 25% |
| **Requirements Engineering** | **Inshan K.C** | **Milan Khatiwada** | **Pankaj Badu** | **Ronak Bastola** |
| Interview plans | 25% | 25% | 25% | 25% |
| Interview findings | 25% | 25% | 25% | 25% |
| Other problem domain research | 25% | 25% | 25% | 25% |
| Academic Literature Review | 25% | 25% | 25% | 25% |
| **Requirements Specification** | **Inshan K.C** | **Milan Khatiwada** | **Pankaj Badu** | **Pankaj Badu** |
| Problem Domain Description | 25% | 25% | 25% | 25% |
| Existing Business Operation | 25% | 25% | 25% | 25% |
| Summary of existing business limitations | 25% | 25% | 25% | 25% |
| Functional Requirements | 25% | 25% | 25% | 25% |
| Performance Requirements | 25% | 25% | 25% | 25% |
| Design and Commercial Constraints | 25% | 25% | 25% | 25% |
| Acceptance Tests | 25% | 25% | 25% | 25% |
| **System Analysis & Design** | **Inshan K.C** | **Milan Khatiwada** | **Pankaj Badu** | **Pankaj Badu** |
| Preliminary Design Stages | 25% | 25% | 25% | 25% |
| Detailed Static System Designs | 25% | 25% | 25% | 25% |
| Detailed Dynamic System Designs | 25% | 25% | 25% | 25% |
| System Database Design | 25% | 25% | 25% | 25% |
| **System Interface Designs** | **Inshan K.C** | **Milan Khatiwada** | **Pankaj Badu** | **Pankaj Badu** |
| Draft Interface Designs | 25% | 25% | 25% | 25% |
| Design Revisions | 25% | 25% | 25% | 25% |
| Heuristic Evaluation | 25% | 25% | 25% | 25% |
| **System Build and Technical Notes** | 25% | 25% | 25% | 25% |
| On-going Testing Methodology | 25% | 25% | 25% | 25% |
| Final System Interface Displays | 25% | 25% | 25% | 25% |
| Technically Difficult Code Sections | 25% | 25% | 25% | 25% |
| **Test Strategy** | **Inshan K.C** | **Milan Khatiwada** | **Pankaj Badu** | **Pankaj Badu** |
| Overview of Test Strategy | 25% | 25% | 25% | 25% |
| Sample Test Results | 25% | 25% | 25% | 25% |
| **System Evaluation** | **Inshan K.C** | **Milan Khatiwada** | **Pankaj Badu** | **Pankaj Badu** |
| Usability Evaluation Strategy | 25% | 25% | 25% | 25% |
| System Pilot Trials | 25% | 25% | 25% | 25% |
| System Trial Results | 25% | 25% | 25% | 25% |
| **Project Conclusions** | **Inshan K.C** | **Milan Khatiwada** | **Pankaj Badu** | **Pankaj Badu** |
|  | 25% | 25% | 25% | 25% |
| **Project Management** | **Inshan K.C** | **Milan Khatiwada** | **Pankaj Badu** | **Pankaj Badu** |
| Project Gantt Chat/WBDS/Activity List | 25% | 25% | 25% | 25% |
| Project Meeting Minutes | 25% | 25% | 25% | 25% |
| Project Quality Plan/Strategy | 25% | 25% | 25% | 25% |

Future scope

After handing the system developed currently it will be modified and it will be enhanced. We can add different features like adding library and leave taking request using the system. Providing this features will help all the users and can be done online.

Test strategy

The testing is done on how the user get the experience of the system. Actual output and expected output are matched in order to know if the code runs as expected.

Sample Test Results

Below are all mentioned test case done for the sytem

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S.N. | Test Case | Process | Expected Output | Actual Output | Result |
| Admin Management | | | | | |
| 1 | Add new admin with all the field or information of the new admin |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

mple Test Results

**6.2.1. Admin testing**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1. Admin management | | | | | |
| S.No. | Test case | Steps | Expected Output | Actual Output | Result |
| 1. | Adding Admin when all form of inputs is filed | * First click the admin tab * Then, click the add admin button * Thirdly, fill up all the information required and submit | Admin should be added | Admin is added | Output is as expected |
| 2. | Adding admin with partially filled form | * First click the admin tab * Then, click the add admin button * Fill up the form partially and continue the further process | Error message should be shown to fill up the form properly | Error message shown | Output is as expected |
| 3. | Editing the admin with proper input procedure | * First click the admin tab * Then, click the edit admin button * Edit the admin information and submit the form | The record of admin should be edited | The admin record is edited | Output is as expected |
| 4. | Delete the admin | * First click the admin tab * Then, click the delete admin button * Confirm the process | Admin should be deleted | Admin deleted | Output is as expected |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1. Module leader management | | | | | |
| S.No. | Test case | Steps | Expected Output | Actual Output | Result |
| 1. | Adding module leader when all form of inputs is filed | * First click the module leader tab * Then, click the add module leader button * Thirdly, fill up all the information required and submit | Module leader should be added | Module leader is added | Output is as expected |
| 2. | Adding module leader with partially filled form | * First click the module leader tab * Then, click the add module leader button * Fill up the form partially and continue the further process | Error message should be shown to fill up the form properly | Error message shown | Output is as expected |
| 3. | Editing the module leader with proper input procedure | * First click the module leader tab * Then, click the edit module leader button * Fill up the form with all required things | The record of module leader should be edited | The module leader record is edited | Output is as expected |
| 4. | Delete the module leader | * First click the module leader tab * Then, click the delete Module leader button * Confirm the process | Module leader should be deleted | Module leader deleted | Output is as expected |
| 5. | Changing the stature of Module leader | * First click the module leader tab * Update the status and submit | Status should be updated | Status updated | Output Is as expected |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1. Admission and student management | | | | | |
| S.No. | Test case | Steps | Expected Output | Actual Output | Result |
| 1. | Adding student when all form of inputs is filed | * First click the student tab * Then, click the add student button * Thirdly, fill up all the information required and submit | student should be added | student is added | Output is as expected |
| 2. | Adding student with partially filled form | * First click the student tab * Then, click the add student button * Fill up the form partially and continue the further process | Error message should be shown to fill up the form properly | Error message shown | Output is as expected |
| 3. | Editing the student information with partially filled form | * First click the student tab * Then, click the edit student button * Edit the record which is to be updated * Submit the file | Error message should be shown to fill up the form properly | Error message is shown | Output is as expected |
| 4. | Editing the student with proper input procedure | * First click the student tab * Then, click the edit student button * Fill up the form with all required things | The record of student should be edited | The student record is edited | Output is as expected |
| 5. | Deleting student information | * First click the student tab * Then, click the student information button * Delete the information and submit | The record of student should be deleted | The record of student is deleted | Output is as expected |
| 6. | Changing the stature of student | * First click the student tab * Update the status and submit | Status should be updated | Status updated | Output Is as expected |
| 7. | Adding documents with invalid files | * First click the student tab * Then, click the student information button * Click the document button * Choose the document file and submit | Error message should be occurred | Error message occurred | Output Is as expected |
| 8. | Adding documents with valid files | * First click the student tab * Then, click the student information button * Click the document button * Choose the document file and submit | Document file should be added | Document file is added | Output Is as expected |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1. Course management | | | | | |
| S.No. | Test case | Steps | Expected Output | Actual Output | Result |
| 1. | Adding Course when all form of inputs is filed | * First click the Course tab * Then, click the Course student button * Thirdly, fill up all the information required and submit | Course should be added | Course is added | Output is as expected |
| 2. | Adding Course with partially filled form | * First click the Course tab * Then, click the add Course button * Fill up the form partially and continue the further process | Error message should be shown to fill up the form properly | Error message shown | Output is as expected |
| 3. | Editing the Course with proper input procedure | * First click the Course tab * Then, click the edit Course button * Fill up the form with all required things | The record of Course should be edited | The Course record is edited | Output is as expected |
| 4. | Deleting Course | * First click the Course tab * Delete the Course and submit | Course should be deleted | Course is deleted | Output is as expected |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1. Module management | | | | | |
| S.No. | Test case | Steps | Expected Output | Actual Output | Result |
| 1. | Adding Module when all form of inputs is filed | * First click the Module tab * Then, click the add Module button * Thirdly, fill up all the information required and submit | Module should be added | Module is added | Output is as expected |
| 2. | Adding Module with partially filled form | * First click the Module tab * Then, click the add Module button * Fill up the form partially and continue the further process | Error message should be shown to fill up the form properly | Error message shown | Output is as expected |
| 3. | Editing the Module with proper input procedure | * First click the Module tab * Then, click the edit Module button * Fill up the form with all required things | The record of Module should be edited | The Module record is edited | Output is as expected |
| 4, | Deleting Module | * First click the Module tab * Delete the Module and submit | Module should be deleted | Module is deleted | Output is as expected |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1. Announcement management | | | | | |
| S.No. | Test case | Steps | Expected Output | Actual Output | Result |
| 1. | Adding Announcement when all form of inputs is filed | * First click the Announcement tab * Then, click the add Announcement button * Thirdly, fill up all the information required and submit | Announcement should be added | Announcement is added | Output is as expected |
| 2. | Adding Announcement with partially filled form | * First click the Announcement tab * Then, click the add Announcement button * Fill up the form partially and continue the further process | Error message should be shown to fill up the form properly | Error message shown | Output is as expected |
| 3. | Editing the Announcement with proper input procedure | * First click the Announcement tab * Then, click the edit Announcement button * Fill up the form with all required things | The record of Announcement should be edited | The Announcement record is edited | Output is as expected |
| 4. | Deleting Announcement | * First click the Announcement tab * Delete the Announcement and submit | Announcement should be deleted | Announcement is deleted | Output is as expected |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1. Time-table management | | | | | | |
| S.No. | Test case | Steps | Expected Output | Actual Output | Result |
| 1. | Adding timetable | * First click the Time-table tab * Then, click the add Time-table button * Thirdly, fill up all the information required and submit | Timetable should be added | Timetable is added | Output is as expected |
| 2. | editing timetable | * First click the Time-table tab * Then, click the edit Time-table button * Thirdly, fill up all the information required and submit | Timetable should be edited | Timetable is edited | Output is as expected |
| 3. | Deleting timetable | * First click the Time-table tab * Then, click the delete Time-table button and confirm the process | Timetable should be deleted | Timetable is deleted | Output is as expected |

**6.2.2. Module leader testing**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1. Module Content management | | | | | |
| S.No. | Test case | Steps | Expected Output | Actual Output | Result |
| 1. | Adding content when all form of inputs are filed | * First click the view module tab * Then, click the view content tab * Thirdly, click the add content button * At last, fill up all the required sections, choose the file and upload | The chosen file must be added added | The chosen file is added | As expected |
| 2. | Adding the content with half-filled sections in form | * First click the view module tab * Then, click the view content tab * Thirdly, click the add content button * At last, fill up all the required sections, choose the file and upload | Error message should be shown | Error message occurred | Output is as expected |
| 3. | Editing the content with fully filled form | * First click the view module tab * Then, click the view content tab * Thirdly, click on edit button * At last edit the information of content and submit | Content should be edited | Content information edited | Output is as expected |
| 4. | Delete the content | * First click the view module tab * Then, click the view content tab * Click on the delete button * Lastly, confirm the process | Content should be deleted | Content is deleted | Output is as expected |
| 5. | Changing the visibility | * First click the view module tab * Then, click the view content tab * Lastly, click on the visibility option | Visibility should be changed | Visibility is changed | Output is as expected |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1. Assignment Management | | | | | |
| S.No. | Test case | Process | Expected Output | Actual Output | Result |
| 1. | Adding assignment when all form of inputs are filed | * First click the view module tab * Then, click the view content tab * Thirdly, click the add assignment button * At last, fill up all the required sections, choose the file and upload | Assignment should be added to the module | Assignment is added | Output is as expected |
| 2. | Adding assignment with half-filled form | * First click the view module tab * Then, click the view content tab * Thirdly, click the add assignment button * At last, fill up all the required sections, choose the file and upload | Error message should be shown | Error message occurred | Output is as expected |
| 3. | edit assignment with filled form | * First click the view module tab * Then, click the view content tab * Thirdly, click the edit assignment button * At last, fill up all the required sections, choose the file and upload | Assignment should be edited | Assignment successfully edited | Output is as expected |
| 4. | Delete the Assignment | * First click the view module tab * Then, click the view content tab * Click on the delete button * Lastly, confirm the process | assignment should be deleted | assignment is deleted | Output is as expected |
| 5. | Changing the visibility | * First click the view module tab * Then, click the view content tab * Lastly, click on the visibility option | Visibility should be changed | Visibility is changed | Output is as expected |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1. Grade Management | | | | | |
| S.No. | Test case | Process | Expected Output | Actual Output | Result |
| 1. | Adding grade | * First click the view module tab * Then, click the assignment tab * Thirdly, click on grades * Choose the obtained grade by student and submit it with your feedback | Grade should be added | Grade is added | Output is as expected |
| 2. | Editing grades | * First click the view module tab * Then, click the assignment tab * Thirdly, click on grades * Selected the grade to be edited * Edit the submitted grade and resubmit it | Grade should be edited | Grade is edited | Output is as expected |
| 3. | deleting grades | * First click the view module tab * Then, click the assignment tab * Thirdly, click on grades * Selected the grade to be deleted * Delete the grade and confirm he process | Grade should be deleted | Grade is deleted | Output is as expected |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1. Attendence Management | | | | | |
| S.No. | Test case | Process | Expected Output | Actual Output | Result |
| 1. | Adding attendance | * First click the view module tab * Then, click the attendance tab * Thirdly, click on the add attendance option * Choose the present student and submit | attendance should be added | Attendance is added | Output is as expected |
| 2. | Sending message for absentee | * First click the view module tab * Choose the module which attendance is to be checked and click view * Click the attendance tab * Click the message button * Write message and send | Message should be sent | Message is send | Output is as expected |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 13 PAT Management | | | | | |
| S.No. | Test case | Process | Expected Output | Actual Output | Result |
| 1. | Sending information to students | * First click the PAT * Then, choose students to send email * Thirdly, fill up the form and send it to students | Email should be sent to chosen students | Email is sent to chosen students | Output is as expected |

* + 1. **Student Testing**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1. Assignment Submission | | | | | |
| S.No. | Test case | Process | Expected Output | Actual Output | Result |
| 1. | Submitting the assignment before deadline | * First click the view module tab * Then, choose the module and view its contents * Thirdly, click the assignment tab * Click the choose file button and upload file * Click the submit button | Assignment should be submitted | Assignment is submitted | Output is as expected |
| 2. | Submitting the assignment after deadline | * First click the view module tab * Then, choose the module and view its contents * Thirdly, click the assignment tab * Click the choose file button and upload file * Click the submit button | Assignment should be submitted in late submission | Assignment is submitted in late submission | Output is as expected |
| 2. | View modules | * Click the view module tab * Then, click the view tab | User should be able to view all module contents | all module contents can be viewed | Output is as expected |
| 3. | Download the module content | * Click the view module tab * Click the content tab * Click on download option | Content should be downloaded | Content is downloaded | Output is as expected |
| 4. | Contact with PAT | * Go to the PAT tab * Click on contact * Write the email and send | Email should be sent | Email is sent | Output is as expected |

**6.2.4. All Users Group**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1. Login | | | | | |
| S.No. | Test case | Process | Expected Output | Actual Output | Result |
| 1. | Login with proper login process | * Fill up all the necessary credentials and submit | Login successful | Login successful | Output is as expected |
| 2. | Login with incorrect login permits | * Fill up the necessary credentials and submit | Login error | Login error | Output is as expected |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1. Logout | | | | | |
| S.No. | Test case | Process | Expected Output | Actual Output | Result |
| 1. | Logout | * Click on the logout button | logout successful | logout successful | Output is as expected |