# ASSIGNMENT COVER SHEET

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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:  **26 June, 2022** | | Extension date: | | | | | Extension agreed by: |
| **Student Names (List each member of the group)**  **Saurab Khadka – 21422060**  **Rohan Niraula – 21422014**  **Sadikshya Baral - 21422015**  **Mukesh Prasad Singh - 21422053**  **Salon Koirala - 21422151** | | | | | | | **Tutor: Mr. 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: | |

Software Engineering Technical Report

Team Ganges

Table of Contents

[1. Introduction 7](#_Toc107165401)

[2. Requirement engineering 8](#_Toc107165402)

[2.1.3.1.1 Record management system 18](#_Toc107165403)

[2.1.3.1.2 Student records/Information portal 20](#_Toc107165404)

[2.1.3.1.3 Student records/Information application (Mobile format) 21](#_Toc107165405)

[2.1.3.2 Academic literature review 24](#_Toc107165406)

[2.1.3.3.1 Example project relevant research 24](#_Toc107165407)

[2.1.3.3.2 Development relevant implications of research 25](#_Toc107165408)

[2.2.2 Functional Requirements 33](#_Toc107165409)

[2.1.3.3.3 Student Records/Information Portal 43](#_Toc107165410)

[2.2.3 Performance Requirements 43](#_Toc107165411)

[2.2.3.1 Records Management 43](#_Toc107165412)

[2.2.3.1.1 Speed 44](#_Toc107165413)

[2.2.3.1.2 Capacity 44](#_Toc107165414)

[2.2.3.1.3 Reliability 44](#_Toc107165415)

[2.2.3.1.4 Usability 44](#_Toc107165416)

[2.2.3.1.5 Accessibility 44](#_Toc107165417)

[2.2.3.2 Student Records/Information Portal 45](#_Toc107165418)

[Capacity 45](#_Toc107165419)

[Reliability 45](#_Toc107165420)

[Usability 45](#_Toc107165421)

[Clear Navigation 45](#_Toc107165422)

[Accessibility 45](#_Toc107165423)

[2.2.4 Design Constraints 46](#_Toc107165424)

[2.2.4.1 Records Management Systems 46](#_Toc107165425)

[2.2.4.2 Student Records/Information Portal 47](#_Toc107165426)

[2.2.5 Commercial Constraints (Total Project) 49](#_Toc107165427)

[3.System Analysis and Design (Record Management System) 50](#_Toc107165428)

[Textual Analysis 50](#_Toc107165429)

[Significant Event Analysis 51](#_Toc107165430)

[Commands, Queries and Constraints 55](#_Toc107165431)

[BON Cluster Charts 62](#_Toc107165432)

[BON Class Charts 64](#_Toc107165433)

[Detailed Dynamic System Designs 69](#_Toc107165434)

[Event Charts 69](#_Toc107165435)

[Dynamic Diagrams 72](#_Toc107165436)

[System Database Design 73](#_Toc107165437)

[Attribute Listings 74](#_Toc107165438)

[4. System Interface Design 77](#_Toc107165439)

[4.1 Draft Interface Design (Record Management) 77](#_Toc107165440)

[4.1.1 Wireframes 77](#_Toc107165441)

[4.1.2 System Navigation Diagram (Record Management) 78](#_Toc107165442)

[Level 0 DFD 79](#_Toc107165443)

[4.1.4 System Activity Event Diagram (Record Management) 84](#_Toc107165444)

[4.2 Design Revisions (Record Management) 90](#_Toc107165445)

[4.4 Draft Interface Design (Student Record/Information Portal) 90](#_Toc107165446)

[4.4.1 Wireframes 90](#_Toc107165447)

[4.4.2 System Navigation Diagram (Student Record/Information Portal) 92](#_Toc107165448)

[4.4.3 System Screen Mock-ups (Student Record/Information Portal) 92](#_Toc107165449)

[4.4.4 System Activity Event Diagram (Student Record/Information Portal) 94](#_Toc107165450)

[4.5 Design Revisions (Student Record/Information Portal) 95](#_Toc107165451)

[5. System Build and Technical Notes 95](#_Toc107165452)

[5.1 Technically Difficult Code Section 95](#_Toc107165453)

[5.1.1 Create Attendance Record 95](#_Toc107165454)

[5.1.2 Take attendance 96](#_Toc107165455)

[6. Test Strategy 96](#_Toc107165456)

[6.1 Overview of Test Strategy 96](#_Toc107165457)

[Black Box Testing: 96](#_Toc107165458)

[White Box Testing: 97](#_Toc107165459)

[Test Results for Record Management System: 97](#_Toc107165460)

[Test Result for Attendance: 97](#_Toc107165461)

[Test Result for Student: 99](#_Toc107165462)

[Test Case for Personal Tutor: 101](#_Toc107165463)

[Test Case for Staff: 103](#_Toc107165464)

[Test Case for Module: 106](#_Toc107165465)

[Test Case for Assignment: 107](#_Toc107165466)

[Test Case for Timetable: 109](#_Toc107165467)

[Test Case for Diaries: 111](#_Toc107165468)

[Test Case for Login: 112](#_Toc107165469)

[Test Case for Report Section: 113](#_Toc107165470)

[Test Cases for Student Portal: 114](#_Toc107165471)

[Test Case for Courses: 114](#_Toc107165472)

[Test Case for Modules: 114](#_Toc107165473)

[Test Case for Assignment: 115](#_Toc107165474)

[Test Case for Personal Tutor: 116](#_Toc107165475)

[Test Case for Timetable: 116](#_Toc107165476)

[Test Case for Report: 117](#_Toc107165477)

[System evaluation Strategy 118](#_Toc107165478)

[7.1 Usability Evaluation Strategy 118](#_Toc107165479)

[7.1.1 Target Audience Selection/ Sample Size Selection 118](#_Toc107165480)

[7.1.2 Trial active task list creation\ design. 118](#_Toc107165481)

[7.2 Initial system pilot trial/ results 118](#_Toc107165482)

[7.2.1 Thinking aloud (observation and results) 118](#_Toc107165483)

[7.2.2 Post Event Walkthrough (Questions) and Results 118](#_Toc107165484)

[7.3 Final System Trial/Results 120](#_Toc107165485)

[7.3.1 Usability Questionnaire Design/ 7.3.2 Final Questionnaire (Used in Trials) 120](#_Toc107165486)

[7.3.2 Questionnaire Results – Implications of Results 120](#_Toc107165487)

[8 Project Conclusion 121](#_Toc107165488)

[9. Project Management 123](#_Toc107165489)

[9.1 Project Gantt Chat/WBDS/Activity List 123](#_Toc107165490)

[9.2 Project Meeting Minutes 123](#_Toc107165491)

[9.3 Project Quality Plan/Strategy 126](#_Toc107165492)

[9.3 Project Work Log 128](#_Toc107165493)

[References: 129](#_Toc107165494)

# Introduction

This is the Project Report of a software proposed for Woodland University Course Management. This report contains various techniques used to find problems in the university's current course management system, solutions to these problems, and the various steps needed to find them. At the end of the report, anyone who has access to this report will have access to current university course management system issues and possible solutions to these issues using detailed step-by-step solutions.

* 1. Project Background

This project addresses issues with Woodland's current course management system. Currently, Woodland University College uses a paper based system to manage courses, but it is difficult to track and dangerous. Therefore, this project will find ways to address and resolve the obstacles of all these problem of this Woodland University. For this project to work best, we need to perform a detailed investigation of the problem. Every solution developed needs to be related to what the customer wants from the system and how the system works. With the help of various techniques, all the problems of the old system are presented in detail and based on this, the appropriate solution is devised. By properly monitoring the performance of the current system and the various unique features that the next system needs to bring that simplify user management without facing the same issues and is even less hassle. Move the project forward. This proposed project can also solve all the problems of the current course management system, store all data in a carefully manage the data management system in a single software. It helps to store all information and online services in single software or website. So overall this project tries to solve all the problems management.

1.2 Project Aims and Objectives

The main goal of this project is to develop software for Woodland University that manages courses of this university in a new computerized system. This software helps to address the issues with paper-based systems that Woodland University College is currently facing.

The purpose of this project is to:

1. Use different techniques to find problem areas in Woodland University's current system.

2) Examine all the problems that the university experiences.

3) Document the obstacles found.

4) Find the necessary solution to these documented problems.

5) Prepare an effective plan and approach for developing the solution set.

6) Thorough and careful development of software sets, taking into account customer needs and expectations.

7) Make sure that the software contains all the required features and that they are all fully functional.

8) Evaluate and test all aspects of the software product.

9) We will confirm and present the correct final system to our customers.

* 1. Project Development Methodology

The methodology used to create this software is agile software development methodology. Agile methodology is a practice that supports the continuous iteration of development and testing throughout the software development lifecycle of a project. The agile software development methodology is one of the easiest and most efficient processes for transforming business needs into software solutions. Agile is a term used to explain software development approaches that involve continuous planning, improvement, collaborative learning in teams, evolutionary development, and early delivery. It calls for flexible responses to change. Agile software development methodology highlight four important keys.

1) Individual and team interactions between tools and processes.

2) Fully documented working software .Clients cooperation and companionship in contract negotiations.

3) Responding to changes includes following a clear plan.

4) We have used this methodology to drive the development of this project and we will be open to any changes the customer makes to the software suite to provide them with the product.

# Requirement engineering

* 1. Elicitation Activities
     1. Interview Plans

To find out the problem domain, an interview with four different interviewees of different roles were interviewed.

All the questions prepared for this interview are documented below.

Questions

* + 1. **Dr. Simon White (Course Leader)**
       1. Could you describe the life cycle of staff and the life cycle of a student?
       2. Will we be getting any further documents regarding the life cycle of students and staff?
       3. Is there any college policy of legal requirement that we must be aware of?
       4. Is there any specific design you want the software to include? For example, a particular theme, use of colours, the font style and so on.
       5. We are assuming not everyone who are accessible to the system can use all the functionalities? Could you briefly explain who can access which functionality and who is prohibited?
       6. Could you tell us about the performance requirements of the software like speed of data, storage capacity and accessibility among users?
       7. Could you explain the record management system, student information portal and corporate website mentioned in the system interface designs?
       8. Can teachers add data from both GUI portion and from the backend?
       9. In terms of all the modules, do you have any of higher priorities? If in case of chance of delay, is there a part we can skip?
       10. What will be the consequence of poor attendance?
       11. Any other information you would like to share with us as a course leader regarding the software and your expectation?
    2. **Mr. Adam Blake (Course Administrator)**
       1. How would you like to organize the students' records?
       2. Could you briefly explain the personal tutor management and report generation area?
       3. What is the diary management system? If you could also elaborate on its functionalities.
       4. What is the role of students in this system? Will they be able to access any functionality in any area or only view the specified data?
       5. Have you had any solutions or any ideas on how to solve the problems of the clerical system currently used?
       6. Is there any other information that you would want to highlight?
    3. **Dr. Raj Singh (Senior lecturer, Module leader, Personal Tutor)**
       1. So, what are your opinions about the system being digitized?
       2. Any functions you would want in the system for the teacher's convenience?
       3. Do you have any suggestions for the software development team?
    4. **Mr. Mark Williams (Existing student)**
       1. What help or services are you expecting from the digitized system?
    5. Interview Findings
       1. Interview Title: Initial Interview with the course leader – Simon White

Interview: Date: 08/05/2022

Duration: 10 minutes

Persons in attendance:

* Saurab
* Sadikshya
* Rohan

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

|  |  |  |
| --- | --- | --- |
| Interviewer | Question Number | Question  Client - Response |
| 1 | Life cycle | |
| Sadikshya | 01 | Could you describe the life cycle of staff and the life cycle of a student?  Students could see the list of the courses. At the time when a student is not enrolled, he/she will have to submit required documents and functions. After the review of all the details submitted by the staff, it is verified. Maybe the student will be required to submit more important documents or will be rejected.  Then a module leader is assigned by the staff and all the follow ups will be conducted based on the system given by the module leaders and the tutors. |
| Saurab | 02 | Will we be getting any further documents regarding the life cycle of students and staff?  At the time, no any additional documents file is prepared regarding anything expect for the ones that is already provided. But in case of any need, it can be communicated through mail. |
| 2 | Policy or legal requirement | |
| Rohan | 03 | Is there any college policy of legal requirement that we must be aware of?  Must be aware of PII data that is Personally Identifiable Information. It will help identify a specific individual. So, this data should not be able to be shared among the students in the system. The PII data must be visible only to the administrative department authorised to this information as the data is also related to information security.  Even if it is not PII data, the academic information should only be visible and accessible to the related person.  From the security point of view, the particular roles like students, tutor, staff should not be given access to any other information except related to them. |
| 3 | Presentation style | |
| Sadikshya | 04 | Is there any specific design you want the software to include? For example, a particular theme, use of colours, the font style and so on.  For the font colour, theme and everything else, Nile can be taken as a reference. It could be used to create the UI of the system.  Other than that, there is no specific requirements. |
| 4 | Functionality accessibility | |
| Saurab | 05 | We are assuming not everyone who are accessible to the system can use all the functionalities? Could you briefly explain who can access which functionality and who is prohibited?  Regarding this topic, a list could be prepared listing the different roles and functionalities in the system. After that, from the client side, a mapping of the privileges and roles can be provided. Based on that matrix, the work can be done.  If after the development of the system some of the access or privileges are to be provoked, the system should provide a function in UI to remove such accesses.  Though at the very beginning it is not necessary to define which role has access to which functionalities. The administrator will handle the functionalities. With this matrix as a base, the development could be executed. |
| 5 | Performance Requirement | |
| Rohan | 06 | Could you tell us about the performance requirements of the software like speed of data, storage capacity and accessibility among users?  At the moment, there are no analytical reports. Hence, if talking about non-functional requirements, it would be good to see the webpage loading within 3 seconds.  As regard to others, the security mapping between the roles and authorised access to the functions is already specified. And about the back end development, backup of the codes should be maintained so that in case of any unfortunate problem, the codes can be easily retrieved. |
| 6 | System Interface Design | |
| Saurab | 07 | Could you explain the record management system, student information portal and corporate website mentioned in the system interface designs?  Put corporate website out of context here.  For the other two, they should be accessible easily. For the enrolment of students as well as better student experience, the student portal should be there. For record management system, as an admin or even as a staff should have their own portal so that they can access to specified functions. |
| Rohan | 08 | Can teachers add data from both GUI portion and from the backend?  Backend is not required for the teachers as it is only accessible to the administrators. |
| 7 | Module priority | |
| Saurab | 09 | In terms of all the modules, do you have any of higher priorities? If in case of chance of delay, is there a part we can skip?  Mobile implementation can be put in lower priority. From the priority point of view, student interface and teacher interface should be of higher priority than others. In fact, more focus on the student system such as being able to send enrolment requests, notifications regarding them.  Basically, delivery of functionalities from the student view should be highly prioritised and gradually can move to tutor and other staff. |
| 8 | Consequences | |
| Sadikshya | 10 | What will be the consequence of poor attendance?  If a student does not have at least 60% of attendance, the admit card will not be provided because of which they would be able to take their exams. |
| 9 | More information | |
| Saurab | 11 | Any other information you would like to share with us as a course leader regarding the software and your expectation?  As a course leader, what is wanted is being able to see what subjects are available in the course and what the modules are. Should have access to the course study materials uploaded by concerned authorities, and if required access to change the materials.  Also, see how the classes are going, their progress and participation is particular classes. Monitor how the assignments are happening and performance of each student. As a course leader, some of the requirements. |

Interview Title: Initial Interview with the course administrator – Adam Blake

Interview: Date: 08/05/2022

Duration: 10 minutes

Persons in attendance:

* Saloon
* Mukesh

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

|  |  |  |  |
| --- | --- | --- | --- |
| Interviewer | Question Number | | Question  Client - Response |
| 1. | Student Record | | |
| Saloon | 1 | How would you like to organize the students' records?  Regarding the student record, student sends an enrollment request, and administrator should view and track all of the students who are enrolled in each specific course, and they will organize them according to his/her age, course, and academic session.  Administrator will also have the option to filtering the student by his/her batch, course, and case. Then, if they have access to Riddick or have approved the following inquiry, then they should follow up with the student if the extra document is requested, and so on. After a student has been enrolled, he or she can view their attendance, academic achievement, and whatever else he wants to see. | |
| 2 | Personal tutor management and report generation | | |
| Mukesh | 2 | Could you briefly explain the personal tutor management and report generation area?  Definitely, there should be the basic functionality. So, students should have the functionality to request a personal tutor if they require one. And the admin should have the functionality to allow or disallow the personal tutor. If they allow it, then only students will have access to the system and follow up.  For the report generation model, it provides the attractive report for all four types of user’s i.e. student, tutor, module leader and admin. So at least few reports should be available for each user type for delivery.  Let’s consider for an admin, a list of course is associated with the student like as attendance report, assignment progress report and so on. So, these are the admin report for the student.  Administrator also want to see the student academic progress that might be one report for them and also see the attendance report for the personal earlier for admin as a whole list of attendance for all the student would be there for the admin. And for the student, they should have the attendance details for the tutor, student record should be allocated to him.  And for the module leader, there should be the progress report of assignment and student participation is showing on report which is related to attendance and so on. At least two three reports for each type of users will be sufficient for this delivery. | |
| 3 | Diary management system | | |
| Mukesh | 3 | What is the diary management system? If you could also elaborate on its functionalities.  In terms of the diary management system, Whatever the functionality that provided is more relevant to, like sending the message or communicating with a particular tutor, then that functionality should be there. In addition to that, if there need to maintain a certain class, let's say some adult classes, that should be communicated to the student. So there should be considering at least two functionalities in short delivery for them. | |
| Saloon | 4 | What is the role of students in this system? Will they be able to Access any functionality in any area or only view the specified data? | |
| 4 | Solution for clerical system | | |
|  | 5 | Have you had any solutions or any ideas on how to solve the problems of the clerical system currently used?  Actually, there is no particular system that adds all the functionalities in one package. But some of the few modules in one system and another few models in another system that will be considerable. And check the course era or apex for some of the functionalities like the course management system, attendance, and exam management system. Some of the functionalities are not there, so investigate some other systems as well inside those kinds of functionalities. And get the reference as well to fill up the gap. | |
| 5 | Information | | |
| Mukesh | 6 | Is there any other information that you would want to highlight? | |

Interview Title: Initial Interview with the Senior lecturer, Module leader, Personal Tutor – Raju Singh

Interview: Date: 08/05/2022

Duration: 10 minutes

Persons in attendance:

* Saurav
* Rohan
* Sadikshya

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

|  |  |  |
| --- | --- | --- |
| Interviewer | Question Number | Question  Client - Response |
| 1 | Opinion | |
| Saurav | 1 | So what are your opinions about the system being digitized?  As a lecturer, they should have the functionality to see the student’s record and his class report. And once a class has been created, and then they get to know which students were present at what time and how much time they spent into the class. Those functionalities must be there.  He wants to see the functionality for the assignment and dragging the progress or the submission, which happened once he read the assignment and published the results. These are the functionalities that he expects as a leader. |
| 2 | Function | |
| Rohan | 2 | Any functions you would want in the system for the teacher's convenience?  As a lecturer, he would love to see the requirements first which has been documented. So that he will give the conformation regarding the requirements and then software development team are following the valid from the methodology then that will be true so which will ensure that whatever the functionality or the working you have been visible or deliverable as a client. And he would have meeting at regular interval of time to help us to find out how the project is going on. |
| 3 | Suggestion | |
| Sadikshya | 3 | Do you have any suggestions for the software development team?  No any specific required file, he would like to give at the moment. So, whatever the file you have that is sufficient for this project. |

Interview Title: Initial Interview with the Existing student – Mark Williams

Interview: Date: 08/05/2022

Duration: 10 minutes

Persons in attendance:

* Saloon

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

|  |  |  |
| --- | --- | --- |
| Interviewer | Question Number | Question  Client - Response |
| 1 | Services expecting | |
| Saloon | 1 | What help or services are you expecting from the digitized system?  As already talks about the functionality, whatever the functionalities have been allocated for the students that need to be organized and provided in the student’s portal. So that experience will be much better. |

2.1.3 Other Problem Domain Research

This part of the project includes an analysis of various existing software systems and related laws that provide additional assistance in understanding other problem areas.

2.1.3.1 Comparable Software System Review

This section explores and compares different software systems available from different sources and shows their strengths and weaknesses.

# 2.1.3.1.1 Record management system

Record management system is simply managing the records of an organization throughout the lifetime. It is the key component of an organization as it provides each and every information which makes it easier to analyze the annual growth, provide space for Improvement, helps in manage the overall organization and so on. It also helps in increasing the accountability and productivity of an organization. For organization like metrology, it is very important to keep record of every data as climate and weather can be predicted but is not constant and always changing. Before all the records were hand written which would be hectic to go through every possible outcome, now due to online record management system, the word load has been declined in huge number. Recording management system is used in each and every sector from school to luxury hotels. Some of the examples of School Record Management System are given below;

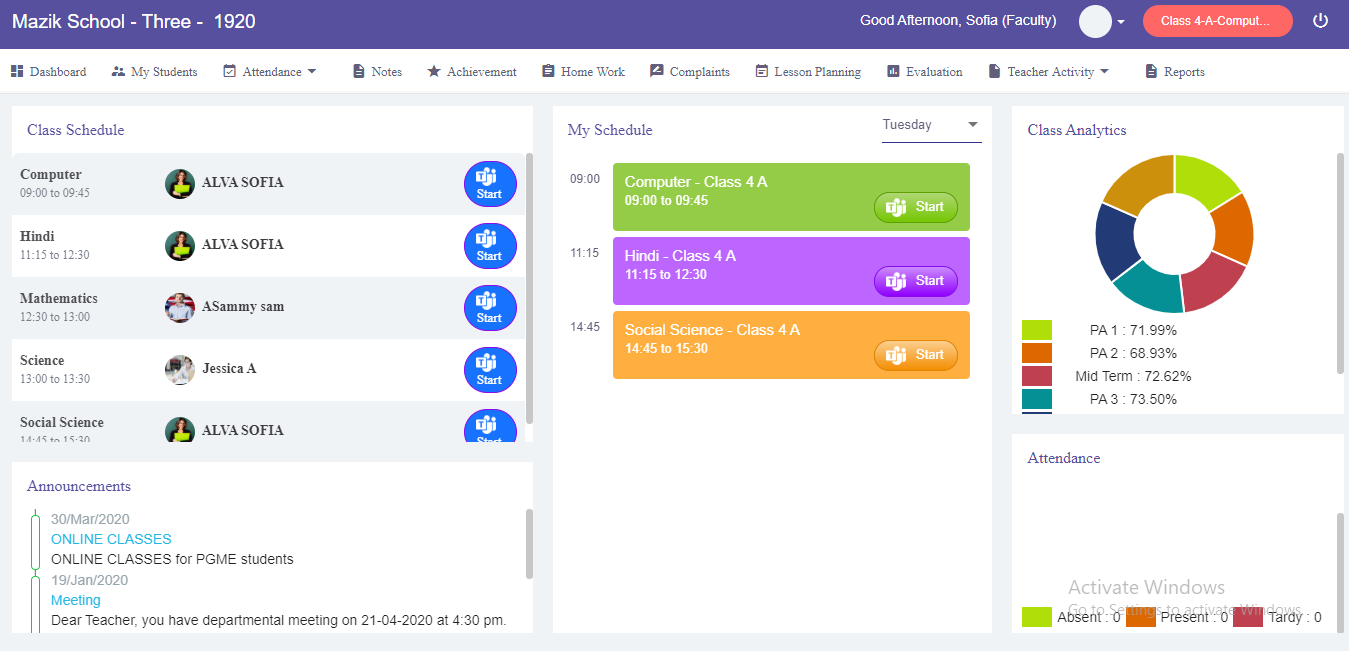
|  |  |
| --- | --- |
| Website | About |
| 1. Vidyalaya | This website is one the best School Management Software in India, which provides management solutions for educational institutes. It provides communication channel between Students, Teachers, Parents also Alumni. 1800+ educational institutes have been using this software.   |  |  | | --- | --- | | **Advantages** | **Disadvantages** | | Multiple campuses can be managed with a single piece of software, requiring less investment in management difficulties. | Does not categorize schooling level; which makes it a bit difficult | | There is access to live demo, which gives you proper investigation and how Vidyalaya works |  | | Reduces work load as it provides information on affiliated institutions |  | |
| 1. Campus EPR | Campus EPR is also an online managing software that provides platform to manage every information, every data systematically. 1000+ Institutes and 100+ countries are affiliated to this software   |  |  | | --- | --- | | **Advantages** | **Disadvantages** | | Website is clean and easily accessible by anybody with minimum information | While going through website, lot of pop ups can be seen | | There is no need for customization; every detail can be obtained easily |  | | Is updated from time to time |  | |
| 1. EConnect-k12 | Econnectk-12 is another EPR system that simplify the school management. It is efficient and secure software that makes school function easy. It is affiliated to 300+ school institutes.   |  |  | | --- | --- | | **Advantages** | **Disadvantages** | | It accessible anytime, anywhere from your computer to your mobile phone | No information on affiliated institutes; which makes it quite difficult to access | | There is a back up tool, which backups all your information if lost |  | | It also is eco-friendly; no use of paper |  | | There is also different tabs for parent and student |  | |

# 2.1.3.1.2 Student records/Information portal

Student’s record system is basically a software that keeps track and manage student data. It makes easier to redeem students’ information within few clicks. It is very beneficial to teachers as it could provide all the task, reports, assignments, and data done by each student. Student record system also organizes information of new registered students, managing fee structures, coordinating examination and so on. Some of the software of student record system are listed below;

**Edmatix**

Edmatix is an EPR software which provides information from preschool to universities in order to compare and provide best outcome and most effective solutions for the doubts of students. The website manages everything from library to home assignment also let parents keep in track to their children’s record.



**Positiveaspects:**

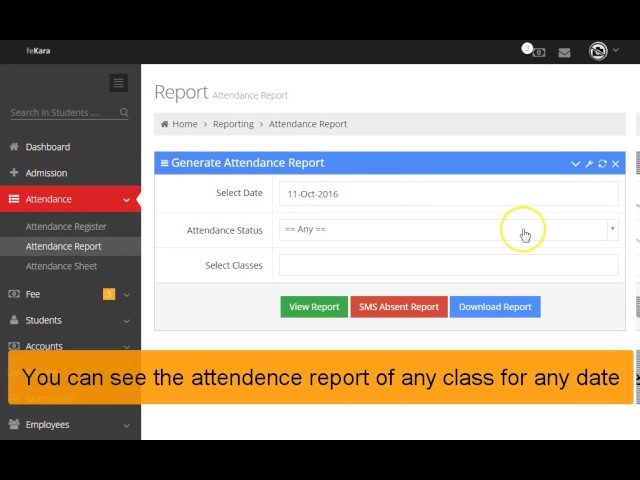
* There are different tabs for Students, Parents and Teachers which can only be accessed respectively
* The website is very well organized, all the information is categorized so it makes easy to understand
* All the information is up to date, and the website is updated from time to time

**Negative aspects:**

* Absence of tool which allows you to immediately contact the institute, you have to first schedule the meeting
* Students are not given much privacy

**feKara**

feKara is also a school software which organizes the students’ life from admission to graduation. It provides the platform to improve students’ life by making everything digital. The website makes easier for teacher to keep in contact to each and every student personally. It is user-friendly and also gives customer service.



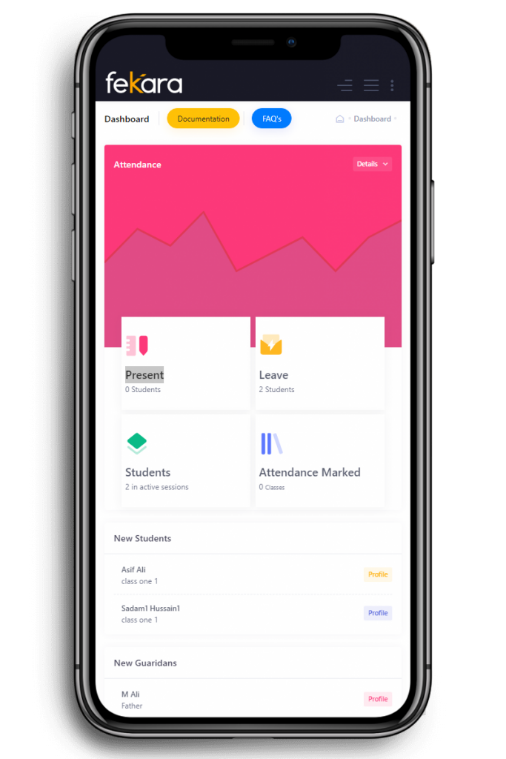
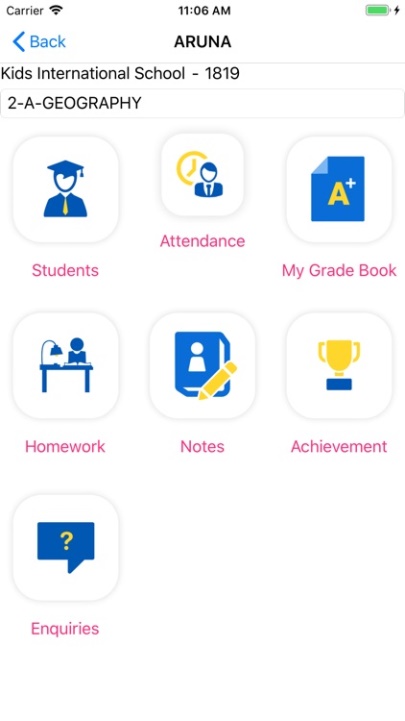
**Positiveaspects:**

* The website has multiple dashboard which makes it easier to use
* It affordable
* It also has different language support hence; it provides better platform for communication
* The website can also be used in mobile

Negative aspects:

* There is no uniqueness in the website

# 2.1.3.1.3 Student records/Information application (Mobile format)

****

Fig; edmatix Fig; feKara

Aside from being extended there is not much difference in web and on mobile phone. Although it is easier to use web, it depends upon one’s preference. Students are to be preferred to use on web as they send more time on using the website, on other hand Parents could use on mobile phone so it can be easily accessible. Regardless, on any devices the effect, the impact is same.

* + - 1. Development Relevant Legislation

2.1.3.2.1 Equality Act:

The Equality Act of 2010 legally protects people against discrimination in the workplace and in society as a whole. We've replaced pre- taboos with a single law that's easier to understand and, in certain cases, enhances protection. It shows the various ways in which it is illegal to treat someone.

Equality Act provisions, effective October 1, 2010:

1) Basis for protection against direct and indirect discrimination, harassment and harm in public, office, work, education, association and traffic. Frames.

2) Modifying the definition of sex reassignment by removing the requirement for medical supervision. 3) Protects people who are believed to have protected characteristics or who are discriminated against because of their relationship with someone with protected characteristics.

4) More obvious protection for nursing mothers.

5) Applying a stable definition of indirect discrimination to all protected establishments.

6) Harmonizing the regulations that allow voluntary and positive behavior.

2.1.3.2.2 General Data Protection Regulations (GDPR):

The supplied software or system for Woodland University College will contain many types of data and information regarding students, academic staff, modules, or persons who are directly or indirectly associated with the college and its new computerized system. Because the information gathered is so sensitive, it must be handled responsibly and in accordance with particular rules and regulations.

Different requirements should be enforced under the Data Protection Act 2018 of the United Kingdom (UK), as stated by the General Data Protection Regulation (GDPR) required in law. According to GOV.UK, 2018, the following are the regulations of privacy and protection of data and information maintained inside the system.

1) Data should be used lawfully and transparently.

2) Data should only be collected for specific and stated purposes.

3) Relevant data should be used sparingly.

4) The information should be used in a suitable and applicable manner.

5) Unnecessary data storage should be limited.

6) The data should only be used within a certain organization's system.

7) Only certified employees of the organization should have access to the organization's key data and information.

2.1.3.2.3 Education Relevant Legislation:

The Education Act 2011 helps teachers raise the standards of their education. It includes new legal authority to help teachers eliminate bad behavior, tackle inefficiencies and improve how schools are held accountable. The provisions of Law are as follows:

1) Permit the search of students for dangerous or prohibited items without the school's consent.

2) Raise restrictions that prohibit schools from notifying students written birth without notice.

3) New advance notice Tax notice Student limitation Crime against teachers.

4) Authorization to facilitate free early childhood education for underprivileged children hardship 2 years old.

5) New school process reform and academy establishment Priority free schools.

6) Refocus Regular school assessment on the four key areas most important to parents.

7) Eligibility for schools to be exempted from regular Ofsted assessments Expand

8) New agency for underperforming schools including.

9) State minister's authority to close these schools.

10) Abolish five independent schools e existing agencies, some of their functions more efficient and report directly to the Secretary of State 44 44 Transferred to law enforcement

# Academic literature review

A literature review is a survey of previously published works on a particular subject. In order to narrow the gap between student, professors and their university, it is the necessity to learn how management system works.

## 2.1.3.3.1 Example project relevant research

Course management system is quite difficult to understand/conduct as they cover a wide distance. Morgan (2003) has stated Course management system as "integrating a set of course content into instructions. “So as to convey and to understand course management system, various survey has been conducted at different University. For example, a survey was conducted in a Hong Kong University where both computer and paper-based options were available. In this University, were about 20000 students and 10 faculties (Architecture, Social studies, Arts, Law, Business and Economics, Dentistry, Education, Engineering, Science and Medicine) where 1/3rd was post graduates and remaining was under graduates. Various survey was carried out such as:

|  |  |
| --- | --- |
| TAble 1: The technologies that students use most | % of students |
| Blog | 62.1% |
| instant messenger | 75.7% |
| email | 95% |
| voice over ip | 26.1% |

|  |  |
| --- | --- |
| table 2: the technology's preferred | Students of different level |
| Blog | Postgraduates |
| rss | Postgraduates |
| wiki | Undergraduates |
| bookmarking | Postgraduates |
| email | Undergraduates |

|  |  |
| --- | --- |
| Table 2: Cms’s students have used | responses |
| moodle | 104 |
| webct | 740 |
| iln | 202 |
|  |  |

And other such investigation was conducted. Although the survey varies from faculty to faculty also gender, this survey led to greater understanding. Through this survey, queries of both students and teachers were considered. It was believed that students were more leaning towards new technologies like Wiki along with Instant messenger. By exploring this survey, understanding student and their perspective on new technologies and course management system were put on consideration.

## Development relevant implications of research

Although this survey has its own drawbacks and not provide accurate information, this has found to be most effective method and has narrowed down the gap between students and professors. This investigation in spite of being conducted in limited area provide the necessary information on how Course Management System is being/have been perceived. Nonetheless, it offers substantial amount of space on how can Course management system can be improved to narrow down the gap between Student and professors. People may not find technology useful simply just because it is present and accessible. Therefore, both parties; Students and Teachers are to be put to consideration and considerate of each other. And there must be some improvements from time to time. Bringing change in Course management system by properly using technology, could bring essential changes in one’s experience. Professors also must realize it’s not just the students who is benefited from Course management system, it facilitates them. On emphasizing it, Course management system would ease the experience of Students, Teachers and their Universities.

2.1.3.4 User Group Questionnaire

2.1.3.4.1 Student Experience Questionnaire

2.1.3.4.1.1 Questionnaire Development

2.1.3.4.1.2 Questionnaire Results

2.1.3.4.1.3 Questionnaire Analysis

🡪Questionnaire development, their results and analysis are all included in following tables:

🡪Current Course Management System

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Strongly Agree | Agree | Average | Disagree | Strongly Disagree |
| You are happy with current course management system. |  |  |  |  |  |
| There is proper management of courses. |  |  |  |  |  |
| You would recommend to use clerical system rather than computerized system. |  |  |  |  |  |
| You are satisfied with reliability of paper based management. |  |  |  |  |  |
| You are satisfied with security of management. |  |  |  |  |  |

🡪New system for course management

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Strongly Agree | Agree | Average | Disagree | Strongly Disagree |
| You believe computerized form is more essential. |  |  |  |  |  |
| You think courses will be managed more efficiently in software. |  |  |  |  |  |
| You think using of software will be more productive for student. |  |  |  |  |  |
| You want software to provide more functions to students. |  |  |  |  |  |
| Good graphics is important for new system. |  |  |  |  |  |
| New system should look professional |  |  |  |  |  |
| Student should get information more easily and quickly. |  |  |  |  |  |

2.1.3.4.2 Staff Experience Questionnaire

2.1.3.4.2.1 Questionnaire Development

2.1.3.4.2.2 Questionnaire Results

2.1.3.4.2.3 Questionnaire Analysis

Questionnaire development, their results and analysis are all included in following table:

🡪Current Course Management System

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Strongly Agree | Agree | Average | Disagree | Strongly Disagree |
| You are happy with current system for managing course. |  |  |  |  |  |
| Clerical system is managing courses properly. |  |  |  |  |  |
| You think use of paper based system is more reliable. |  |  |  |  |  |
| You are satisfied with current system. |  |  |  |  |  |
| You would recommend others to use clerical system. |  |  |  |  |  |

🡪New system for course management

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Strongly Agree | Agree | Average | Disagree | Strongly Disagree |
| Computerized system is more important in today’s world. |  |  |  |  |  |
| You believe there will be effective management in software. |  |  |  |  |  |
| Use of computerized system will be more beneficial. |  |  |  |  |  |
| Data and Information will be more secured. |  |  |  |  |  |
| Staff should have more important functions. |  |  |  |  |  |
| Staff should be able to provide information’s to student. |  |  |  |  |  |
| Software should be more professional |  |  |  |  |  |

2.2 Requirements Specification

2.2.1 Problem Domain Description

2.2.1.1 Existing Business Operation

In the way of developing the new course system, there should be a clear description, investigation, and overview of current company activities before building a new system. Advantages and disadvantages of the current system of that institution could be identified by studying its history or current system, that is an important task of software engineering. The domain knowledge identification is aided by a thorough analysis of the previous system in the requirement specification. Current system of this course management system includes the process for student life cycle, personal tutorial life cycle and employee life cycle.

2.2.1.1.1 Student Life-Cycle

The Student Life Cycle outlines the phases that students take from pre-enrolment to retention and engagement, and then to completion and comment achievement. Understanding the Student Life Cycle can help those responsible for instilling a sense of belonging and providing student support services, as well as encouraging ideal learning environments. In fact, effectively controlling that life cycle is critical to generating outstanding learning outcomes.

The Student Life Cycle is a frame of reference that depicts our students' experience. It's a 'living architecture' which can be modified and modified to meet the evolving needs of every incoming group of students, unique teaching / learning methodologies, integrated educational support service provides, and also other reflective teaching activities.

Educational leaders use the Student Life Cycle framework to assist in part of the planning, forecast, and scheduling, as well as to assist build new approach to boost 1st engagement and achievement of students. Finally, knowing the Student Life Cycle process will also help instructors to plan and construct holistic students’ academic activities, while bearing in view that a number of attributes should influence and influence the process andimpact a student's perseverance and success, which will contribute significantly in assisting them in settling in as new pupils and finding a common bond. There are eight different phases into the student-life cycle as shown in figure below-

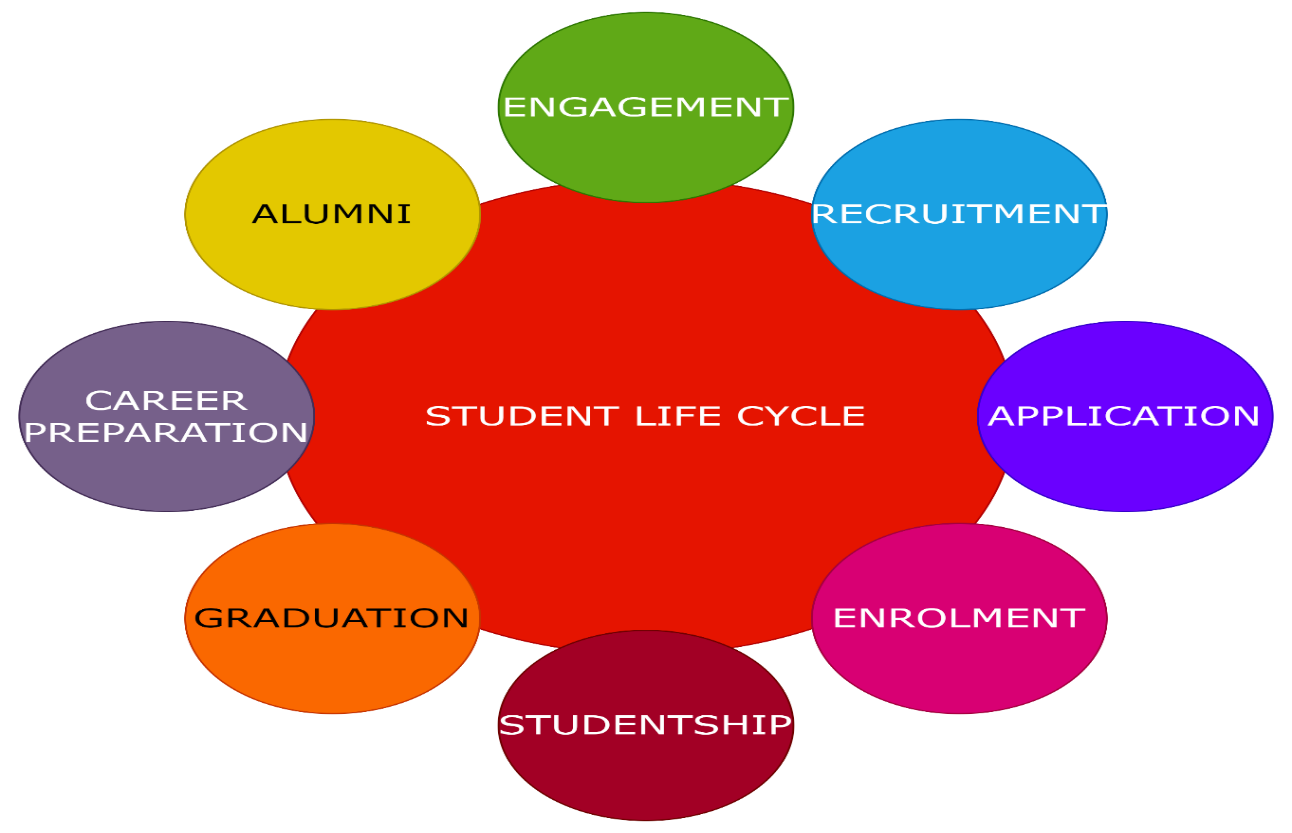


Fig- Student life cycle model

1. Engagement:All through the students’ experiences, there is interaction to future, existing, as well as former classmates.
2. Recruitment: Student should be recruitment into the Woodlands University College (WUC).
3. Application: This phase helps the student to determine apply into the Woodlands University College (WUC). In this phase also student is informed of the result of application.
4. Enrolment: This phase of life cycle is to Enrol the student into the Woodlands University College (WUC) at the first year of the college.
5. Studentship:Actions which 'change' students academically and generally, allowing students can evolve as active 'public academics.
6. Graduation: This phase of life cycle helps the student to complete the graduation into the Woodlands University College (WUC).
7. Career Preparation: The actions which engage learners in considering about, developing skills in, and planning to your comment professions take place across all 3 stages of the life cycle.
8. Alumni:Participation in events which foster commitment Woodlands University College (WUC)both in university and beyond graduation.

2.2.1.1.2 Personal Tutorial Life-Cycle

Personal tutor plays the vital role to helps the student to get the success and support their journey. at the university. They guide the student and make friendly environment among the students and university. They are the mediator of collage and student as well.

Tutor provides all the support that needed to the student Like as learning materials, information about exam, assignment and so on. And they also listen or find out the student and give the proper solution for them. And encourage the student to engaged into the different types of new development work. Students felt attached with the university to see themselves as physically and socially integrated seem to be more likely to stay around and develop. Mainly its life cycle work on eight stages as we seen given below figure.

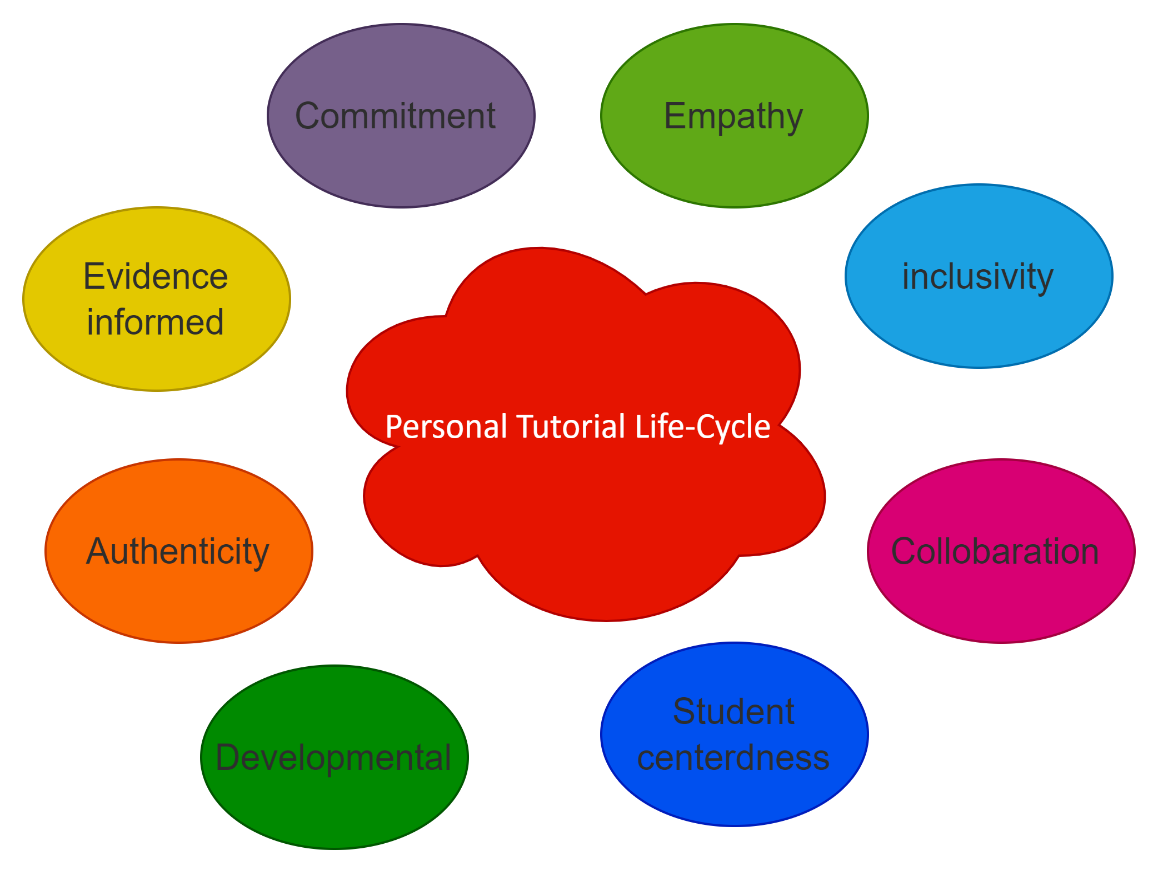


Figure- Personal Tutorial Life Cycle

1. Empathy: In this phase of life cycle, students have lot of problem at the university. But tutor understand the problems of student and help them to find out the solution. And also encourage the student to labour hard to find out the new things.
2. Inclusivity: In this phase of life, tutor helps the student providing learning materials, assignment and so on for practice and get the knowledge.
3. Collaboration: This phase of life cycle helps the student make familiar environment to the university. Tutor is the mediator between university and the student.
4. Student Centredness: This phase of life, tutor makes the seat planning for the student while exam. And helps the student sit according to the seat planning.
5. Development: This phase of life, tutor encourage the student to develop new things. And also provide material to the student to develop anything new.
6. Authenticity: In this phase of life, tutor inspire the student to be loyal and engagement toward the university. And inspire you to lead the team for any project.
7. Evidence Informed: In this phase of life cycle, tutor also have to find out the evidence while any student is doing unusual activities into the college or university.
8. Commitment: In this phase of life cycle, commitment very important for the tutor. They are responsible for the all the activities are happening into the university.

2,2.1.1.3 Employee Life Cycle

The employee life cycle holds the relation among the staff and their company.This starts with the attraction period, where employee is unknown with the company. They still get to know the company, and continuing unless they resign by becoming alumnus, and also as we called him, joyful quitter. There are six phases into the employee life cycle that are shown into the figure-

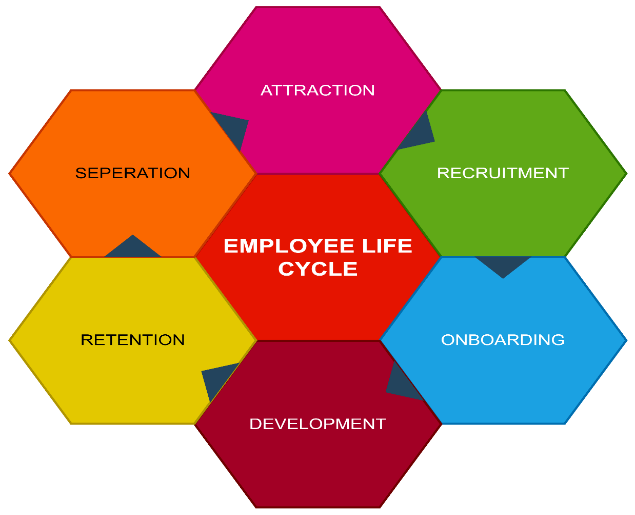


Fig- Employee life cycle

1. Attraction: This is the initial step of the employee lifecycle. This relates to the organization strategy for attracting new employees. A business must choose how that will recruit and retain employees. Until an organization can figure out is how to identify employees, it must first determine what it wants to be remembered for and how it will differentiate itself from the competition. Corporate attraction is influenced by factors such as brand awareness, company culture, and employee perks and benefits.
2. Recruitment: The employee recruiting stage is the second stage of the employee life cycle, during which they search out and hire the greatest personnel for your company. Hiring can happen when a current post becomes empty or when a fresh one is formed. The most approaches give a positive employee experience, encourage collaborative selecting based on defined guidelines and procedures, and give actionable information which would be leveraged to enhance recruitment outcomes.
3. Onboarding:Onboarding is the next stage in the employee life cycle model. The onboarding stage, which begins if you have hired talented employees, is crucial for rapidly and smoothly acclimating the new employees toward the company culture and performs of his new position. New staff address further components of particular role during the integration phase, identifying the mindsets, skills, abilities, and actions required to succeed inside the firm.
4. Development: This is the fourth stages of employee life cycle model. During such a time, you should start to routinely promote career prospects within their staff, and that will also assist them improve their knowledge and give them with a future career path inside the organization.
5. Retention: This is the fifth employee life cycle model. This is where should concentrate their efforts upon retaining their best personnel and ensure that they must be pleased and engaged in their roles into the organization. In this stage, the organization environment and its impact. If your company's environment is terrible, you'll undoubtedly have just a significant attrition rate, which means you have to hire employees on a frequent basis. Enhancing your recruitment phase is an excellent method to mitigate such risk and enhance career longevity and happiness for their team.
6. Separation:This is the sixth stages of life cycle model. Usually, staff will reach the end of their ELC at some point, either through resignation, steady job, or own issues. It's vital to approach the exit process as seriously as the new employee orientation and to handle it with the same level of strategy. While one of your associates quits, and has an impact on those.

2.2.1.2 Summary of existing business limitations requiring resolution and existing strengths.

While working on this project, we find out the lot of limitation int the current system. Some of the limitations of current system are given below-

1. Security issues: In existing system, all the data and information keep into the paper. But we face the main problem in security of data and information. We may lose or mishandled the document.
2. Lengthy issues: In paper-based system, data and information keep into paper but regular entry of data it becomes lengthy.so we have very difficult to find the data and information from the paper.
3. Limitation of storage: It is extremely difficult to save data and information in a paper-based system. It will take a lot of room to preserve information on paper, and we'll have to utilize multiple pages as the amount of information to be stored grows. so that it could be rapidly retrieved, that is problematic in a paper-based system because we may have not enough stored the data while needed.
4. Announcement issues: Any notification, such as a result or information, is posted on the existing system's notice board. As a result, there is more time spent getting the announcement, and the results may be missed on the notice board.
5. Attendance issues: As we keep the attendance into the paper. student or teacher may forget the take the attendance. It consumes more time and difficult to store the data.
6. Transportation of document:In a paper-based system, document transit is highly dangerous because it must be physically carried from one location to another. Online-based systems can quickly control document transit by email, which is very convenient for many of us as well as for law enforcement and other government agencies.
7. Editing of document: In existing system, while keeping the data or information into the paper we may done some mistakes. So, there is very difficult to edit those data in paper-based system. We have to re-write the data beginning from new paper. and for this small mistake we have to repeat all the data.
8. High cost: One of the main limitations of a paper-based system is that it is more costly than an online system. In this system, multiple materials are required to manage paper documents, that will result in increased costs over time. However, because all papers are controlled in one software in an online-based system, it is far less expensive than a paper-based system.

## 2.2.2 Functional Requirements

A functional requirement is a definition of behavior between inputs and outputs that describes a function of a system or its component. Calculations, technical details, data manipulation, processing, and other specific functionality that define what a system is expected to perform are examples of functional requirements. Behavioral requirements are used to describe all of the scenarios in which the system applies the functional requirements, which are represented in use cases.

**2.2.2.1.1 Records Management Systems**

A records management system manages the administration of records for an organization throughout the records-life cycle. This facilitates the methodical and efficient management of record deletion, deletion, and related business transactions.It is a precise, simple, and safe system that includes features such as monitoring and updating student records.The various records management systems are listed below.

* **Student Records managements**

1. All information about the students is recorded in an organized manner in this management.
2. With just a few clicks, users may locate and retrieve student information.
3. Students can manage their personal information in Student Records, and academic and administrative staff can view student information and perform student administration tasks.

The student records operation list is provided below.

|  |  |  |  |
| --- | --- | --- | --- |
| Operation | Admins | Staff | Student |
| Create | Create a new record | No access | No access |
| Amend | Add a record to the achieve list | No access | No access |
| Archive | Transfer a record to the achieve | No access | No access |
| Display | Displays a list of all records, with all information displayed when an item is clicked | Students name and contact details are visible. | They can see their own name and rank. |
| Assign | Assign a course to a student | No access | No access |

2.2.2 Functional Requirements

A functional requirement is a definition of behavior between inputs and outputs that describes a function of a system or its component. Calculations, technical details, data manipulation, processing, and other specific functionality that define what a system is expected to perform are examples of functional requirements. Behavioral requirements are used to describe all of the scenarios in which the system applies the functional requirements, which are represented in use cases.

**2.2.2.1.1 Records Management Systems**

A records management system manages the administration of records for an organization throughout the records-life cycle. This facilitates the methodical and efficient management of record deletion, deletion, and related business transactions.It is a precise, simple, and safe system that includes features such as monitoring and updating student records.The various records management systems are listed below.

* **Student Records Managements**

1. All information about the students is recorded in an organized manner in this management.
2. With just a few clicks, users may locate and retrieve student information.
3. Students can manage their personal information in Student Records, and academic and administrative staff can view student information and perform student administration tasks.

Access right:

This table contains information about the users who have access to the system.

|  |  |  |  |
| --- | --- | --- | --- |
| **Operation** | **Admin** | **Staff** | **Student** |
| Create | Create a new record | No access | No access |
| Amend | Add a record to the achieve list | No access | No access |
| Archive | Transfer a record to the achieve | No access | No access |
| Display | Displays a list of all records, with all information displayed when an item is clicked | Students name and contact details are visible. | They can see their own name and rank. |
| Assign | Assign a course to a student | No access | No access |

A simplified entry list for a student record is shown below.

|  |  |  |
| --- | --- | --- |
| **Name of entity** | **Type/format** | **Notes** |
| Student\_id | NUMBER(10) | Primary key for students |
| Address | VARCHAR2(15) | Address of student |
| Record status | Enum(live, dormant) | The student's current status is recorded |
| Course\_id | NUMBER(2) | Unique key for course |
| Date\_of\_birth | NUMBER(12) | Date of birth of student |
| Full\_name | VARCHAR2 | Full name of student |
| Contact | VARCHAR10 | Contract of the student |

* **Staff Records Managements**

1. For better operations, the staff management should be offered in a computerized or digital format.
2. The admins should have full access; staff should not be able to delete or manage student records, but they should have access to the system.

Access right

This table contains information about the users who have access to the system. Users are Admin, staff, student.

|  |  |  |  |
| --- | --- | --- | --- |
| **Operation** | **Admin** | **Staff** | **Student** |
| Create | Access | No access | No access |
| Amend | Access | No access | No access |
| Achieve | Access | No access | No access |
| Display | Access | Access | No access |
| Assign | Access | No access | No access |

A simplified entry list for a staff record is shown below.

|  |  |  |
| --- | --- | --- |
| **Name of entity** | **Type/format** | **Notes** |
| Staff\_id | NUMBER(2) | Primary key for staff |
| Full\_name | String (limited to 30 characters) | Full name of the staff |
| Address | VARCHAR2(17) | Address of staff |
| Contact | NUMBER(10) | Phone number of staff |
| Course\_name | VARCHAR2(10) | Name of course |
| Module\_id | NUMBER(2) | Unique key for module |

* **Course Records Management**

A course management system is a set of software tools that allows students to engage with one other in an online context.It establishes an atmosphere where course modules, students, and faculty members can obtain news, publish notices, and communicate urgent information to students.

Access right;

This table contains information about the users who have access to the system. Three types of user are Admin, staff, student.

|  |  |  |  |
| --- | --- | --- | --- |
| **Operation** | **Admin** | **Staff** | **Student** |
| Create | Access | No access | No access |
| Structure | Access | No access | No access |
| Amend | Access | No access | No access |
| Display | Access | Access | Access |
| Delete | Access | No access | No access |
| Achieve | Access | No access | No access |

A simplified entry list for a course record is shown below.

|  |  |  |
| --- | --- | --- |
| **Name of entity** | **Type/format** | **Notes** |
| Course\_id | NUMBER(2) | Unique key for course |
| Course\_name | VARCHAR2(10) | Name of course |
| Module\_id | NUMBER(2) | Unique key for module |

* **Module Management System**

The Module Management System is an online component that aids in the management and administration of modules, as well as ensuring regulatory compliance.It establishes a framework for maintaining transparency between course modules, students, and administration management.

The Management Module serves as the SAFE blueprint's command and control module. The security support infrastructure is housed in this module.

Access Right;

This table contains information about the users who have access to the system. Three types of user are Admin, staff, student.

|  |  |  |  |
| --- | --- | --- | --- |
| **Operation** | **Admin** | **Staff** | **Student** |
| Create | Access | No access | No access |
| Amend | Access | No access | No access |
| Achieve | Access | No access | No access |
| Display | Access | Access | Access |
| Delete | Access | No access | No access |
| Assign | Access | No access | No access |

A simplified entry list for a module management system is shown below.

|  |  |  |
| --- | --- | --- |
| **Name of entity** | **Type/format** | **Notes** |
| Module\_id | NUMBER(2) | Unique key for module |
| Module\_name | VARCHAR2(10) | Name of course |
| Staff\_id | NUMBER(2) | Primary key for staff |

* **Assignment Management System**

A course module can use an assignment management system to schedule tasks and allocate them to the appropriate students. Additionally, students are given deadlines, and once they submit their work, the course module evaluates it.

Each subject's assignments and projects can be viewed separately. It is also simple for the course module to assign topics to students from the system's pre-defined topics.

Before submitting assignments, students can complete and revise them as needed. The works cannot be modified after they have been submitted.

Access right;

This table contains information about the users who have access to the system. Three types of user are Admin, staff, student.

|  |  |  |  |
| --- | --- | --- | --- |
| **Operation** | **Admin** | **Staff** | **Student** |
| Create | Access | No access | No access |
| Amend | Access | No access | No access |
| Achieve | Access | No access | No access |
| Display | Access | Access | Access |
| Delete | Access | No access | No access |
| Assign | Access | Access | No access |
| Mark/Grade | Access | Access | No access |

A simplified entry list for assignment management system is shown below.

|  |  |  |
| --- | --- | --- |
| **Name of entity** | **Type/format** | **Notes** |
| Id |  |  |
| Assignment\_name | VARCHAR2(15) | Name of assignment |
| Module\_id | NUMBER(2) | Unique key for module |
| Initialized\_date | DATE | Initialized date of assignment |
| Deadline\_Information | VARCHAR(13) | Deadline information of assignment |
| Student\_id | NUMBER(10) | Primary key for student |
| Staff\_id | NUMBER(2) | Primary key for staff |

* **Attendance Records Management**

The attendance management system tracks employees' and students' actual working hours, time off, and login and logout times in order to monitor their attendance.

Access right:

This table contains information about the users who have access to the system. Three types of user are Admin, staff, student.

|  |  |  |  |
| --- | --- | --- | --- |
| **Operation** | **Admin** | **Staff** | **Student** |
| Create | Access | No access | No access |
| Amend | Access | Access | No access |
| Archive | Access | No access | No access |
| Display | Access | Access | No access |
| Delete | Access | No access | Access |
| Monitor | Access | No access | No access |
| Action Poor Attendance | Access | No access | No access |

A simplified entry list for attendance record is shown below.

|  |  |  |
| --- | --- | --- |
| **Name of entity** | **Type/format** | **Notes** |
| Attendance\_percent | PERCENTAGE | Attendance in percent |
| Student\_id | NUMBER(10) | Primary key for student |
| Student\_name | VARCHAR2 | Full name of student |
| Year | DATE | Date when attendance record |

* **Personal Tutor Management**

The Personal Tutor serves as an intermediary between the personal and academic realms, establishing a friendly and nonjudgmental relationship of trust with their tutee and providing a context for the formation of peer relationships through group tutorials, both of which benefit and enhance students' academic and personal development**.**

Before sending a student to a more specialized source of support, it is critical that tutees understand the function of the Personal Tutor and the limitations of what a Tutor can do**.**

Access right;

This table contains information about the users who have access to the system. Three types of users are Admin, staff, student.

|  |  |  |  |
| --- | --- | --- | --- |
| **Operation** | **Admin** | **Staff** | **Student** |
| Create | Access | No access | No access |
| Amend | Access | No access | No access |
| Display | Access | No access | Access |
| Assign | Access | No access | No access |

A simplified entry list for apersonal tutor managementis shown below.

|  |  |  |
| --- | --- | --- |
| **Name of entity** | **Type/format** | **Notes** |
| Staff\_id | NUMBER(2) | Primary key for staff |
| Module\_id | NUMBER(2) | Unique key for module |
| Student\_id | NUMBER(10) | Primary key for student |

* **Timetable Management System**

A timetable management is a type of schedule that specifies the dates and times of specified events. It could also mean;

Timetable, a table where students, personnel and other resources can be co-ordinated

Time horizon**,** a future date or time when certain processes will be reviewed or deemed to be completed.

Access right;

This table contains information about the users who have access to the system. Three type of users are Admin, Staff, Student.

|  |  |  |  |
| --- | --- | --- | --- |
| **Operation** | **Admin** | **Staff** | **Student** |
| Create | Access | No access | No access |
| View | Access | Access | Access |
| Amend | Access | No access | No access |
| Archive | Access | No access | No access |
| Delete | Access | No access | No access |

A simplified entry list for a timetable management system is shown below.

|  |  |  |
| --- | --- | --- |
| **Name of entity** | **Type/format** | **Notes** |
| Course\_id | NUMBER(2) | Unique key for course |
| Day | Enum | Day for timetable management |
| Time | NUMBER(15) | Time for table management |
| Module\_id | NUMBER(2) | Unique key for module |

* **Diary Management System**

Diary management entails keeping track of both your diary and your daily activities. It will use in following ways;

Make a diary in advance.

Make a copy of your diary.

Gather as much information as possible.

Access right;

This table contains information about the users who have access to the system. Three types of user are Admin, student, staff.

|  |  |  |  |
| --- | --- | --- | --- |
| **Operation** | **Admin** | **Staff** | **Student** |
| Create | Access | No access | No access |
| View | Access | Access | Access |
| Amend | Access | Access | No access |
| Prompt | Access | No access | No access |
| Initiate Automated Action | Access | No access | No access |

A simplified entry list for a diary management system is shown below.

|  |  |  |
| --- | --- | --- |
| **Name of entity** | **Type/format** | **Notes** |
| Id | NUMBER(2) | Unique identifier for diary |
| Message | VARCHAR2(15) | Message for diary management |
| Schedule | VARCHAR2(10) | Schedule for diary management |
| Pictures | VARCHAR2(15) | Picture for diary management |
| Student\_id | NUMBER(10) | Primary key for student |

* **Report Generation/Management**

The process of extracting the data you need from a database, structuring it, and exporting it into reports is known as report generation. It supplies decision-makers with useful information as well as references.

The structure selects reports that are valuable to it from the library's whole collection. The status of favourite can be assigned to these reports. Favourites are then conveniently displayed in the navigation menu for quick access.

Access right;

This table contains information about the users who have access to the system. Three types of User are Admin,student, staff.

|  |  |  |  |
| --- | --- | --- | --- |
| **Operation** | **Admin** | **Staff** | **Student** |
| Create | Access | No access | No access |
| View | Access | Access | Access |
| Print | Access | No access | No access |

A simplified entry list for a report management system is shown below.

|  |  |  |
| --- | --- | --- |
| **Name of entity** | **Type/format** | **Notes** |
| Staff\_id | NUMBER(2) | Primary key of staff |
| Student\_id | NUMBER(10) | Primary key for student |

## 2.1.3.3.3 Student Records/Information Portal

Student Records are records, files, documents, and other materials maintained by a university or college that contain information directly related to students and are accessible to other professional personnel to aid in the instruction, guidance, and educational advancement of student**.**

The technical requirements for the Student Records/Information Portal are shown below.

|  |  |
| --- | --- |
| Requirements | Description |
| Home /login page | This page will contain a login portal requiring username and password. |
| Student Information page | This page will contains detail information of student. |
| Course | This page will display all courses and view Module content, Reading and resources, About module, Module activities and Assignment work. |
| Messages page | Through this page student direct message to the course module regarding the problem faced in course or assignment work. |
| Mark/Grade | This page will display grade of current courses. |
| About us | Woodlands University College’s details, description and history will be found in this page. |
| Sign out | Clicking the bottom of sign out portal. |

## 2.2.3 Performance Requirements

Performance requirements are usually a set of criteria that specify how things should perform or the standards that they must meet under certain conditions. This differs from prescriptive requirements, which explain exactly how something should be done.

Briefing documentation, performance-based specifications, output-based specifications, legislation, and so on may all contain performance criteria.

## 2.2.3.1 Records Management

The supervision and administration of digital or paper records, regardless of format, is known as records management (RM). The creation, receiving, maintenance, usage, and disposal of records are all part of records management.

## 2.2.3.1.1 Speed

The following are some major characteristics that influence the speed of our system:

The time it takes to process a batch of records is known as throughput. At any given time, this system can process a hundred thousand records.

In a data or record management system, optimal speed means that records are processed quickly, efficiently, and smoothly

## 2.2.3.1.2 Capacity

The capacity of the record management system refers to the amount of workload that the system can handle without causing strain or damage to the system.

It displays the amount of data that can be stored in this system as well as the number of operations that can be performed concurrently.

a. The maximum number of records that the system can store. This system has a storage capacity of 50 TB.

b. The total number of target users who can simultaneously access the online system. This system can support up to 100 concurrent users.

## 2.2.3.1.3 Reliability

A reliable record is one whose contents may be relied upon in the course of later transactions or activities as a complete and accurate portrayal of the transactions, activities, or facts to which they witness.

The most important requirement for any record management system to function in the future is the dependability of its data management system. It should be efficient and maintain a constant level of performance under pressure.

## 2.2.3.1.4 Usability

The record management system must be simple to operate. It should not take a large amount of labor to operate, and the system's operation should not be overly complicated or require extensive training. It is the rate at which users are able to understand this system, which includes data recording, update, insert, delete, and system backup.

It is possible to locate, retrieve, present, and understand a usable record.

## 2.2.3.1.5 Accessibility

The capacity to use and benefit from a system or entity is referred to as accessibility. Access controls the complete information lifecycle, allowing you to keep track of your entire application. Information that is easily accessible, regardless of its shape, structure, or presentation.

## 2.2.3.2 Student Records/Information Portal

**Speed**

Different factors influence the speed of the student/information portal. Due to the various areas on the student/information portal, such as the login page, message page, information page, and others, processing takes time depending on the number of users. For example, if there are only 1 or 2 users, the average login time is only 30-40 seconds.

It may take roughly 10-15 seconds if the number of connection requests does not reach 100 because the server is likely busy, and it will only take a few seconds if there are just one or two users.

### Capacity

The total number of students accessing the student portal and the student portal reaction, i.e. the speed with which the student portal responds as the number of students grows. A student portal's capacity is determined by two factors: the number of students and the rise in material. As the number of users grows, the server may lag, hang, or slow down.

### Reliability

Student portal reliability is a quality that functions consistently and dependably. Out of a total of 730 hours per month, our student portal system will be operational for 700 hours. Our student portal has a backup plan and strategy in place in case of system failure. We have to configure a physical network for the backup plan, as well as install the student portal, backup, security, and storage policies, as well as their requirements.

### Usability

With basic knowledge and training, this decides how easy it is to use a student portal. There are various ways to assess your student portal's usability.

* Performance

The speed with which potential clients access the student portal determines its performance. The number of pupils and the substance of the page determine the speed. It takes around one minute to load a server if the number of students is larger than 100, and 3035 seconds to load a server if one student is greater than 100.

### Clear Navigation

We have a range of options to bring you to the desired page because our student portal is student-centric. For example, we can see a navigation bar on our page with login, information, course, mark/grade, report page, and so on. When a user clicks on that menu to see information about a certain student, it displays that student's details.

### Accessibility

It refers to the page where students enter their username and password to gain access to a learning organization's programs and other learning materials. Students can also use the student portal to tell Accessibility Services about future tests, quizzes, or examinations so that a place can be reserved for them.

After logging in, the Student Dashboard screen will appear, displaying your portal activities.

### 2.2.4 Design Constraints

## 2.2.4.1 Records Management Systems

Design constraints are the non-fundamental requirement of the system. It describes how the system should be design for the record management system. Client prescribed himself what they need into the system. Here client play the vital role to build up the system. We execute almost functionality that are prescribed by the client.

Typical design constraint most of the system should be included are-

1. Target Operation System:

For the record management system, System will run in all the operating system which are existing in the market like as window, Linux, Apple Mac OS, Android and so on.

1. Distributed or local architecture:

The system of record management components can run on different devices in a spread or remote design, and then they can communicate with one another across a networking to reach their aims. And The system architecture should be local as possible and for university, includes the extra storage and copies.

1. Required hardware requirements:

For the hardware requirement for the student record management assess into any common hardware included into the own laptop/computer and other devices like mobile phones can be used to perform this software, and its functions properly on all systems without causing any disturbance.

1. Front-end graphic styles:

For the font-end graphic style, client was not giving any idea for the system like its color, graphic style and so on to design the system. so, we design the record management system with our own ideas to make the system attractive. And the system also contains the buttons, rout menu and so on.

1. Programming languages to be used:

For the system design, client not prescribed the certain language to design the system. but we use the React, Mango DB and Node JS to make the system attractive.

1. Application packages:

For manage the data record and store the data into the system which are indicated for the application of the system.

1. Development standard:

Software should be handed within the deadline that given from the client. And all the functionality which are prescribed by the client should be in the system. and all the resource are implemented within the budget.

1. Design approaches:

For the design approaches, client is conjugate the OOP concept into the requirement file so it design the system has hardly executed the OOP ideas.

## 2.2.4.2 Student Records/Information Portal

Design constraint are the non-functional requirement of the system. It describes about how the system should be design for the student records/information portal. Client himself describes all the required portal or information for the student. Here client plays the vital role to build up the system. After design constraint description from the client, we have executed almost design constraint into the system.

Typical design constraint most of the system should be included are-

1. Target Operation System:

This software will run in all the operating system which are existing in the market like as window, Linux, Apple Mac OS, Android and so on.

1. Distributed or local architecture:

The system's components can run on different devices in a spread or remote design, and then they can communicate with one another across a networking to reach their aims. And The system architecture should be local as possible and for university, includes the extra storage and copies.

1. Required hardware requirements:

For the hardware requirement for the student record/ information portal assess into any common hardware included into the own laptop/computer and other devices like mobile phones can be used to perform this software, and its functions properly on all systems without causing any disturbance.

1. Front-end graphic styles:

For the font-end graphic style, client was not giving any proper color, graphic style and so on to design the system. So, we design the student record/ information portal with our own ideas to make the system attractive.

1. Programming languages to be used:

For the system design, client not prescribed the certain language to design the system. but we use the React, Mongo DB, Node JS, and MERN to make the system attractive.

1. Application packages:

For manage the data record and store the data into the system which are indicated for the application of the system.

1. Development standard:

Software should be handed within the deadline that given from the client. And all the functionality which is prescribed by the client should be in the system. And all the resource is implemented within the budget.

1. Design approaches:

For the design approaches, client is conjugate the OOP concept into the requirement file so it design the system has hardly executed the OOP ideas.

2.2.6 Acceptance Tests (Total Project)

Record management system

|  |  |  |  |
| --- | --- | --- | --- |
| S. N | Record | Acceptance Test | Explanation |
| 1 | Student | Student Id, Address, Record status, course id, DOB, Name, Contact | This all record should be implemented into the student record management. |
| 2 | Staff | Staff Id, Name, Address, Contact, Course name, Module ID | These all staff record to be implemented for the staff record management. |
| 3 | Course | Course Id, name | All record to for the course management. |
| 4 | Module | Module id, module name, staff id | All record should be assessable for the module management. |
| 5 | Assignment | Assignment name, id, Deadline student id, staff id | For assignment, this all record to be recorded. |
| 6 | Attendance | Attendance percentage, student id, name, year | For the attendance, these all record to be implemented. |
| 7 | Personal Tutor | Staff id, student id, module id | For Personal tutor, this all record to be recorded. |
| 8 | Diary Management | Id, massage, schedule, picture, student id | For diary management, all the tested information to be recorded |
| 9 | Report Generation | Staff id, student id | For the Report generation, staff id and student id have to be recorded to find the report of staff and student. |

Student Records/Information Portal

|  |  |  |  |
| --- | --- | --- | --- |
| S.N | Function | Acceptance Test | Explanation |
| 1 | Home/ login page | Page successfully login and open home page | While we access the system, firstly we have to login into the system then only home page will open. |
| 2 | Student information page | Successfully open student information page | While clicking on Student information page then students’ information should be there. |
| 3 | Course | Open Successfully course portal | While clicking on course, all the courses will be shown. |
| 4 | Message page | Open Successfully massage portal | While clicking on message portal, massage box will be open. And u can massage the administrator |
| 5 | Grade | Open Successfully Grade portal | While clicking grade portal, then grade of subject will be shown. |
| 6 | About Us | Open Successfully About us portal | While clicking on about us, then description about the intuition is shown. |
| 7 | Sign out | Sign out successfully from the system | While clicking on sing out then you can easily sign out from the system. |

Above all system, we work hard on it. And as per as client interview, we implement all the function into the system and make the system more attractive. The entire acceptance test gives us positive preview. So, we assume that the system run properly and give better output.

## 2.2.5 Commercial Constraints (Total Project)

**Budget Estimated: Around** $30,000,000

The budget for each element is shown in the table below.

Assignment Information:

Project duration in total: 5 week

Cost per hour: $1000 per person

Members of the group: 5 members

|  |  |  |  |
| --- | --- | --- | --- |
| SECTION | WEEKS | HOURS WEEK PER PERSON | TOTAL COST |
| Requirements Specification | 1 | 4 | $12,00,000 |
| Design and Analysis | 1 | 5 | $13,50,000 |
| System Interface Designs | 1 |  | $12,00,000 |
| Requirement Engineering | 1 | 6 | $2,50,000 |
| Problem Domain | 1 | 5 | $32,00,000 |

|  |  |
| --- | --- |
| REASON | COST |
| Software | $5,00,000 |
| Office cost | $11,44,000 |
| Total build cost | $2,00,000 |
|  |  |
| Other expenses (20%) | $30,00,000 |
| Profit margin (30%) | $60,00,000 |
| Cost of the entire project | $2,400,000 |

# 3.System Analysis and Design (Record Management System)

Here, the current system is analysed and with the help of the requirement specification that is created, better procedures and methods are designed to make the system more effective. Here, the system will implement the Business Object Notation (BON).

* 1. Preliminary Design Stages

## Textual Analysis

Textual analysis will model the possible classes of the system as well as the behaviours of those classes. This is done with the help of the documents provided and the necessary researches.

|  |  |
| --- | --- |
| Candidate Class | Class Responsibilities/Behaviours |
| Student | create\_student, amend\_student, archive\_student, display\_student, assign\_student |
| Staff | create\_staff, amend\_staff, archive\_staff, display\_staff,  assign\_staff |
| Course | create\_course\_structure, amend\_course, display\_course, delete\_course, assign\_course |
| Module | create\_module, amend\_module, delete\_module, archive\_module, display\_module, assign\_module |
| Assignment | create\_assignment, amend\_ assignment, delete\_ assignment, archive\_ assignment, display\_ assignment, assign\_ assignment, mark |
| Attendance | create\_attendance, amend\_ attendance, archive\_ attendance, monitor, display\_ attendance, action\_poor\_attendance |
| PersonalTutor | create\_tutor, amend\_tutor, assign\_tutor, display\_tutor |
| Timetable | create\_timetable, amend\_timetable, delete\_timetable, archive\_timetable, display\_timetable |
| Diary | create\_diary, amend\_diary, display\_diary, prompt\_diary, initiate\_automated\_action |
| Report | create\_report, display\_report, print\_report |

## Significant Event Analysis

Here, the possible classes in the system is identified with the help of significant events in the system. In other words, actions that will result to change in the system state.

|  |  |  |
| --- | --- | --- |
| Event | Performer | Candidate Attributes |
| Login to the system | Administrator | User Id  Password |
| Log out of the system | Administrator | - |
|  | | |
| Add student record | Administrator | Id  First Name  Middle Name  Surname  Address (Term Time)  Address (Non- term if diff)  Contact email  Course ID  Year  Section  Tutor Id  Information link |
| Update/Amend student record | Administrator |
| Archive student record | Administrator |
| View student record | Administrator, Student | Id |
| Assign student | Administrator |
|  | | |
| Add staff record | Administrator | Id  First Name  Middle Name  Surname  Address  Contact  Email  Role  Module Id |
| Update/Amend staff record | Administrator |
| Archive staff record | Administrator |
| View staff record | Administrator, Staff | Id |
| Assign staff | Administrator |
|  | | |
| Add course structure | Administrator | Id  Course Name  Module Id |
| Update/Amend course | Administrator |
| Archive course | Administrator |
| View course | Administrator, Staff, Student | Id |
| Delete course | Administrator |
|  | | |
| Add module | Administrator | Id  Module Name  Staff Id  Course Id |
| Update/Amend module | Administrator |
| Delete module | Administrator |
| Archive module | Administrator |
| View module | Administrator, Staff, Student |
| Assign module | Administrator |
|  | | |
| Add assignment | Administrator | Id  Assignment Name  Module Id  Initialized Date  Deadline  Information  Submission link |
| Update/Amend Assignment | Administrator |
| Delete Assignment | Administrator |
| Archive Assignment | Administrator |
| View Assignment | Administrator, Staff, Student | Id |
| Assign assignment | Administrator, | Id |
| Mark | Administrator, Staff | Id  Student Id |
|  | | |
| Generate attendance record | Administrator | Attendance percent  Student Id  Student Name  Year |
| Update/Amend attendance | Administrator, Staff |
| Archive attendance | Administrator |
| View attendance | Administrator, Staff |
| Action poor attendance | Administrator |
| Monitor | Administrator | - |
|  | | |
| Add personal tutor | Administrator | Staff Id  Module Id  Student Id |
| Update/Amend tutor | Administrator |
| View tutor | Administrator, Student |
| Assign tutor | Administrator | Staff Id  Student Id |
|  | | |
| Add timetable | Administrator | Course Id  Day  Time  Module Id |
| Update/Amend timetable | Administrator |
| Archive timetable | Administrator |
| Delete timetable | Administrator | Course Id |
| View timetable | Administrator, Staff, Student | Course Id |
|  | | |
| Generate report | Administrator | Staff Id  Student Id |
| View report | Administrator, Staff, Student |
| Print report | Administrator | - |
|  | | |
| Create diary | Administrator | Id  Message  Schedule  Student Id |
| Update/Amend diary | Administrator, Staff |
| Prompt | Administrator |
| View diary | Administrator, Staff, Student | Id |
| Initiated Automated Action | Administrator |

## Commands, Queries and Constraints

**Login**

|  |  |  |  |
| --- | --- | --- | --- |
| CLASS | LOGIN | | PART 1/1 |
| TYPE OF OBJECT  A login object in the system. | | Author - Saurab Khadka  Created – 9 May, 2022 | |
| Queries | | User Id, Password, Status | |
| Commands | | Login | |
| Constraints | | * The user Id and password must be correct. * There is three possible user types: administrator, staff and student. | |

**Student**

|  |  |  |  |
| --- | --- | --- | --- |
| CLASS | STUDENT | | PART 1/1 |
| TYPE OF OBJECT  A student record object in the system. | | Author - Saurab Khadka  Created – 9 May, 2022 | |
| Queries | | Id, Full Name, Address, Contact, DOB, Year, Section, Information link | |
| Commands | | Create, Amend, Archive, Display, Assign | |
| Constraints | | * Id must be unique. * DOB should of valid date. * Should not be able to enrol to more than one course. * Contact must be unique. * Administrator can access every command. * Student can only view his/her record. | |

**Staff**

|  |  |  |  |
| --- | --- | --- | --- |
| CLASS | STAFF | | PART 1/1 |
| TYPE OF OBJECT  A staff record object in the system. | | Author - Saurab Khadka  Created – 9 May, 2022 | |
| Queries | | Id, Full Name, Address, Contact | |
| Commands | | Create, Amend, Archive, Display, Assign | |
| Constraints | | * Id must be unique. * Contact must be unique. * Administrator can access every command. * Staff can only view their own record. | |

**Course**

|  |  |  |  |
| --- | --- | --- | --- |
| CLASS | COURSE | | PART 1/1 |
| TYPE OF OBJECT  A course record object in the system. | | Author - Saurab Khadka  Created – 9 May, 2022 | |
| Queries | | Id, Course Name | |
| Commands | | Create, Amend, Archive, Display, Delete | |
| Constraints | | * Id must be unique. * Administrator can access every command. * Staff and student can only view the course Id and name. | |

**Module**

|  |  |  |  |
| --- | --- | --- | --- |
| CLASS | MODULE | | PART 1/1 |
| TYPE OF OBJECT  A module management object in the system. | | Author - Saurab Khadka  Created – 9 May, 2022 | |
| Queries | | Id, Module Name | |
| Commands | | Create, Amend, Archive, Display, Delete | |
| Constraints | | * Id must be unique. * Administrator can access every command. * Staff and student can only view the module Id and name. | |

**Assignment**

|  |  |  |  |
| --- | --- | --- | --- |
| CLASS | ASSIGNMENT | | PART 1/1 |
| TYPE OF OBJECT  An assignment management object in the system. | | Author - Saurab Khadka  Created – 9 May, 2022 | |
| Queries | | Id, Assignment Name, Initialized date, Deadline, Information, Submission link | |
| Commands | | Create, Amend, Archive, Display, Delete, Assign, Mark | |
| Constraints | | * Id must be unique. * Initialized date and deadline must be valid. * Administrator can access every command. * Staff can assign and mark. * Student can only view the assignment queries. | |

**Attendance**

|  |  |  |  |
| --- | --- | --- | --- |
| CLASS | ATTENDANCE | | PART 1/1 |
| TYPE OF OBJECT  An attendance record object in the system. | | Author - Saurab Khadka  Created – 9 May, 2022 | |
| Queries | | Attendance percent, Year | |
| Commands | | Create, Amend, Archive, Display, Monitor, Action Poor Attendance | |
| Constraints | | * Students with attendance percent less than 40% shall face appropriate actions. * Administrator can access every command. | |

**Personal Tutor**

|  |  |  |  |
| --- | --- | --- | --- |
| CLASS | PERSONAL\_TUTOR | | PART 1/1 |
| TYPE OF OBJECT  A personal tutor management object in the system. | | Author - Saurab Khadka  Created – 9 May, 2022 | |
| Queries | | Staff Id | |
| Commands | | Create, Amend, Assign, Display | |
| Constraints | | * Id must be unique. * Contact must be unique * Administrator can access every command. * Student can only view the personal tutor queries. | |

**Timetable**

|  |  |  |  |
| --- | --- | --- | --- |
| CLASS | TIMETABLE | | PART 1/1 |
| TYPE OF OBJECT  A timetable management object in the system. | | Author - Saurab Khadka  Created – 9 May, 2022 | |
| Queries | | Day, Time | |
| Commands | | Create, Amend, Archive, Display, Delete | |
| Constraints | | * Id must be unique. * Administrator can access every command. * Staff and student can only view the timetable. | |

**Report Generation**

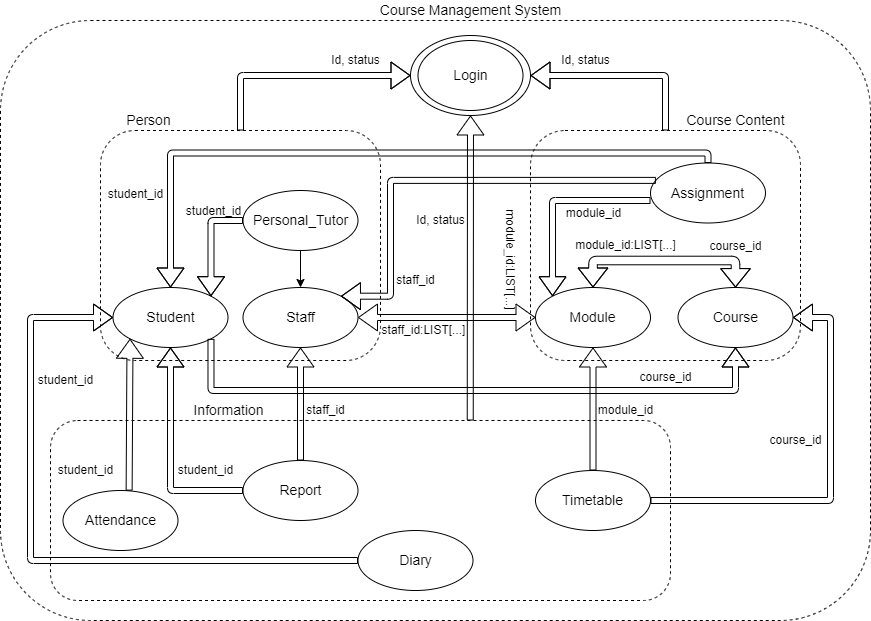
|  |  |  |  |
| --- | --- | --- | --- |
| CLASS | REPORT\_GENERATION | | PART 1/1 |
| TYPE OF OBJECT  A report generation object in the system. | | Author - Saurab Khadka  Created – 9 May, 2022 | |
| Queries | |  | |
| Commands | | Create, Display, Print | |
| Constraints | | * Administrator can access every command. * Staff and student can only view their own report. | |

**Diary**

|  |  |  |  |
| --- | --- | --- | --- |
| CLASS | DIARY | | PART 1/1 |
| TYPE OF OBJECT  A diary management object in the system. | | Author - Saurab Khadka  Created – 9 May, 2022 | |
| Queries | | Id, Message, Schedule, Pictures | |
| Commands | | Create, Amend, Display, Prompt, Initiate Automated Action. | |
| Constraints | | * Administrator can access every command. * Staff can update and view diary. * Student can only view the diary. | |

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

The diagram below is a draft of the system architecture made with the help of the Business Object Notation (BON).



BON System Architecture Diagram

* + - * 1. BON System Chart

|  |  |  |  |
| --- | --- | --- | --- |
| SYSTEM | COURSE MANAGEMENT SYSTEM | | PART 1/1 |
| PURPOSE  Develop an effective system for course management. | | Author - Saurab Khadka  Created – 10 May, 2022 | |
| Cluster | | Description | |
| Person | | Records data of people of different roles in the university along with corresponding access to make changes. | |
| Course Content | | Records the different courses available in the university along with more detailed information such as modules included and assignments assigned. | |
| Information | | Records data about the timetable, attendance, information diary as well generated report about staff and student. | |

## BON Cluster Charts

|  |  |  |  |
| --- | --- | --- | --- |
| CLUSTER | PERSON | | PART 1/1 |
| PURPOSE  Record data of people of different roles. | | Author - Saurab Khadka  Created – 10 May, 2022 | |
| Cluster Components | | Description | |
| STUDENT | | Records all the necessary data about students as well as makes changes in the records as necessary. | |
| STAFF | | Records all the necessary data about staff as well as makes changes in the records as necessary. | |
| PERSONAL\_TUTOR | | Records all the necessary data about personal tutors as well as makes changes in the records as necessary. | |

|  |  |  |  |
| --- | --- | --- | --- |
| CLUSTER | COURSE CONTENT | | PART 1/1 |
| PURPOSE  Record data of courses and other course details. | | Author - Saurab Khadka  Created – 10 May, 2022 | |
| Cluster Components | | Description | |
| COURSE | | Records all the information about the courses available in the university as well as makes in the records changes as necessary. | |
| MODULE | | Records all the about the modules taught in the university as well as makes changes in the records as necessary. | |
| PERSONAL\_TUTOR | | Records all the necessary data about personal tutors as well as makes changes in the records as necessary. | |

|  |  |  |  |
| --- | --- | --- | --- |
| CLUSTER | INFORMATION | | PART 1/1 |
| PURPOSE  Record data other important details such as schedule, attendance and so on. | | Author - Saurab Khadka  Created – 10 May, 2022 | |
| Cluster Components | | Description | |
| ATTENDANCE | | Records data about student’s attendance as well as makes changes in the records as necessary. | |
| TIMETABLE | | Records the timetable of the courses as well as makes changes in the records as necessary. | |
| REPORT | | Generates report about all the record of student and staff of the university. | |
| DIARY | | Keeps record of any necessary schedules or any other information provided to the students. | |

## BON Class Charts

|  |  |  |  |
| --- | --- | --- | --- |
| CLASS | LOGIN | | PART 1/1 |
| TYPE OF OBJECT  A login object in the system. | | Author – Saurab Khadka  Created – 9 May, 2022 | |
| Queries | | User Id, Password, Status | |
| Commands | | Login | |
| Constraints | | * The user Id and password must be correct. * There is three possible user types: administrator, staff and student. | |

|  |  |  |  |
| --- | --- | --- | --- |
| CLASS | STUDENT | | PART 1/1 |
| TYPE OF OBJECT  A student record object in the system. | | Cluster – Person  Created – 10 May, 2022 | |
| Queries | | Id, Full Name, Address, Contact, DOB, Year, Section, Information link | |
| Commands | | Create, Amend, Archive, Display, Assign | |
| Constraints | | * Id must be unique. * DOB should of valid date. * Should not be able to enrol to more than one course. * Contact must be unique. * Administrator can access every command. * Student can only view his/her record. | |

|  |  |  |  |
| --- | --- | --- | --- |
| CLASS | STAFF | | PART 1/1 |
| TYPE OF OBJECT  A staff record object in the system. | | Cluster – Person  Created – 10 May, 2022 | |
| Queries | | Id, Full Name, Address, Contact | |
| Commands | | Create, Amend, Archive, Display, Assign | |
| Constraints | | * Id must be unique. * Contact must be unique. * Administrator can access every command. * Staff can only view their own record. | |

|  |  |  |  |
| --- | --- | --- | --- |
| CLASS | PERSONAL\_TUTOR | | PART 1/1 |
| TYPE OF OBJECT  A personal tutor management object in the system. | | Cluster – Person  Created – 10 May, 2022 | |
| Queries | | Staff Id | |
| Inherits from | | Staff | |
| Constraints | | * Id must be unique. * Contact must be unique * Administrator can access every command. * Student can only view the personal tutor queries. | |

|  |  |  |  |
| --- | --- | --- | --- |
| CLASS | COURSE | | PART 1/1 |
| TYPE OF OBJECT  A course record object in the system. | | Cluster – Course Content  Created – 10 May, 2022 | |
| Queries | | Id, Course Name | |
| Commands | | Create, Amend, Archive, Display, Delete | |
| Constraints | | * Id must be unique. * Administrator can access every command. * Staff and student can only view the course Id and name. | |

|  |  |  |  |
| --- | --- | --- | --- |
| CLASS | MODULE | | PART 1/1 |
| TYPE OF OBJECT  A module management object in the system. | | Cluster – Course Content  Created – 10 May, 2022 | |
| Queries | | Id, Module Name | |
| Commands | | Create, Amend, Archive, Display, Delete | |
| Constraints | | * Id must be unique. * Administrator can access every command. * Staff and student can only view the module Id and name. | |

|  |  |  |  |
| --- | --- | --- | --- |
| CLASS | ASSIGNMENT | | PART 1/1 |
| TYPE OF OBJECT  An assignment management object in the system. | | Cluster – Course Content  Created – 10 May, 2022 | |
| Queries | | Id, Assignment Name, Initialized date, Deadline, Information, Submission link | |
| Commands | | Create, Amend, Archive, Display, Delete, Assign, Mark | |
| Constraints | | * Id must be unique. * Initialized date and deadline must be valid. * Administrator can access every command. * Staff can assign and mark. * Student can only view the assignment queries. | |

|  |  |  |  |
| --- | --- | --- | --- |
| CLASS | ATTENDANCE | | PART 1/1 |
| TYPE OF OBJECT  An attendance record object in the system. | | Cluster – Information  Created – 10 May, 2022 | |
| Queries | | Attendance percent, Year | |
| Commands | | Create, Amend, Archive, Display, Monitor, Action Poor Attendance | |
| Constraints | | * Students with attendance percent less than 40% shall face appropriate actions. * Administrator can access every command. | |

|  |  |  |  |
| --- | --- | --- | --- |
| CLASS | TIMETABLE | | PART 1/1 |
| TYPE OF OBJECT  A timetable management object in the system. | | Cluster – Course Content  Created – 10 May, 2022 | |
| Queries | | Day, Time | |
| Commands | | Create, Amend, Archive, Display, Delete | |
| Constraints | | * Id must be unique. * Administrator can access every command. * Staff and student can only view the timetable. | |

|  |  |  |  |
| --- | --- | --- | --- |
| CLASS | REPORT\_GENERATION | | PART 1/1 |
| TYPE OF OBJECT  A report generation object in the system. | | Cluster – Course Content  Created – 10 May, 2022 | |
| Queries | |  | |
| Commands | | Create, Display, Print | |
| Constraints | | * Administrator can access every command. * Staff and student can only view their own report. | |

|  |  |  |  |
| --- | --- | --- | --- |
| CLASS | DIARY | | PART 1/1 |
| TYPE OF OBJECT  A diary management object in the system. | | Cluster – Course Content  Created – 10 May, 2022 | |
| Queries | | Id, Message, Schedule | |
| Commands | | Create, Amend, Display, Prompt, Initiate Automated Action. | |
| Constraints | | * Administrator can access every command. * Staff can only update and view diary. * Student can only view the diary. | |

## Detailed Dynamic System Designs

### Event Charts

The interaction between objects is caused due to the incoming events or external events in the system. In event charts, the interesting and representative enough events are listed along with the objects involved in those events.

|  |  |  |  |
| --- | --- | --- | --- |
| **EVENTS** | **STUDENT\_RECORD** | | **PART 1/1** |
| **COMMENT**  Selected incoming/external events triggering illustrative types of behaviour | | Created – 10 May, 2022 | |
| **External (Incoming)** | | **Involved Object Types** | |
| Request to login to the system. | | LOGIN | |
| Request to add a new student record | | STUDENT, COURSE | |
| Request to edit a student record | | STUDENT, COURSE | |
| Request to archive the student record | | STUDENT, COURSE | |
| Request to view the details of a student | | STUDENT, COURSE | |

* + - * 1. Object Creation Charts

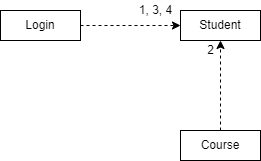
|  |  |  |  |
| --- | --- | --- | --- |
| **EVENTS** | **COURSE\_MANAGEMENT\_SYSTEM** | | **PART 1/1** |
| **COMMENT**  List of classes creating objects in the system. | | Created – 10 May, 2022 | |
| **Class** | | **Creates instances of** | |
| LOGIN | | - | |
| STUDENT | | COURSE | |
| STAFF | | MODULE | |
| COURSE | | MODULE | |
| MODULE | | STAFF, COURSE | |
| ASSIGNMENT | | MODULE, STUDENT | |
| ATTENDANCE | | STUDENT | |
| PERSONAL\_TUTOR | | STAFF, STUDENT | |
| TIMETABLE | | COURSE, MODULE | |
| DIARY | | STUDENT, STAFF | |
| REPORT\_GENERATION | | STUDENT, STAFF | |

* + - * 1. System Scenario Charts

|  |  |  |  |
| --- | --- | --- | --- |
| **SCENARIOS** | **STUDENT\_RECORD** | | **PART 1/1** |
| **COMMENT**  Selected some fascinating scenarios to show important kinds of system behaviour. | | Created – 10 May, 2022 | |
| **Check if the Id and password is correct:**  After filling up the user Id and Password, these field are checked from the database. And if it is, logged into the system, if not, cannot login. | | | |
| **Create a student record:**  All the necessary field asked to store the data is to be filled properly. All the entered data is stored in the database. One field is taken from course class. | | | |
| **Edit the record just created:**  That particular record is chosen All the data recorded about that student is displayed and necessary updates are made and re-updated into the database. | | | |
| **Archive the created record:**  The record in the database is then archived along with all other records previously kept. | | | |
| **View the record:**  Record can be viewed from the database about that particular student chosen. Every details stored in that database is displayed. | | | |

### Dynamic Diagrams

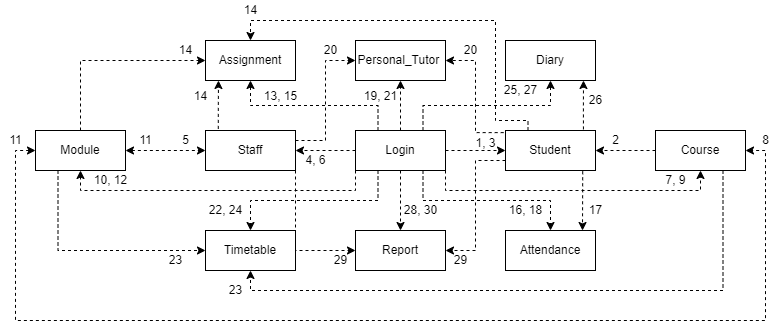
|  |
| --- |
| Scenario:   * + - 1. Open student record       2. Ask course id from course       3. Include course id in student record       4. Display student record |



BON Dynamic Diagram of Student record

Here is an overall scenario of the course management system as well as its dynamic diagram.

|  |
| --- |
| Scenario:   1. Log into the system and open student. 2. Access course Id from course. 3. Display student. 4. Check the staff. 5. Import module Id from module. 6. Display staff. 7. Open course. 8. Access module list. 9. Display course. 10. Open module. 11. Import staff Id from staff and course Id from course. 12. Display module. 13. Open assignment. 14. Access student Id, staff Id and module Id. 15. Display assignment. 16. Select attendance. 17. Obtain student Id. 18. Show attendance. 19. Open personal tutor. 20. Inherit from staff and import student Id. 21. Show tutor. 22. Check timetable. 23. Access course Id and module Id. 24. Show timetable. 25. Open diary. 26. Access section from student. 27. Display diary. 28. Select report generation. 29. Access student Id and staff Id. 30. Display report. |

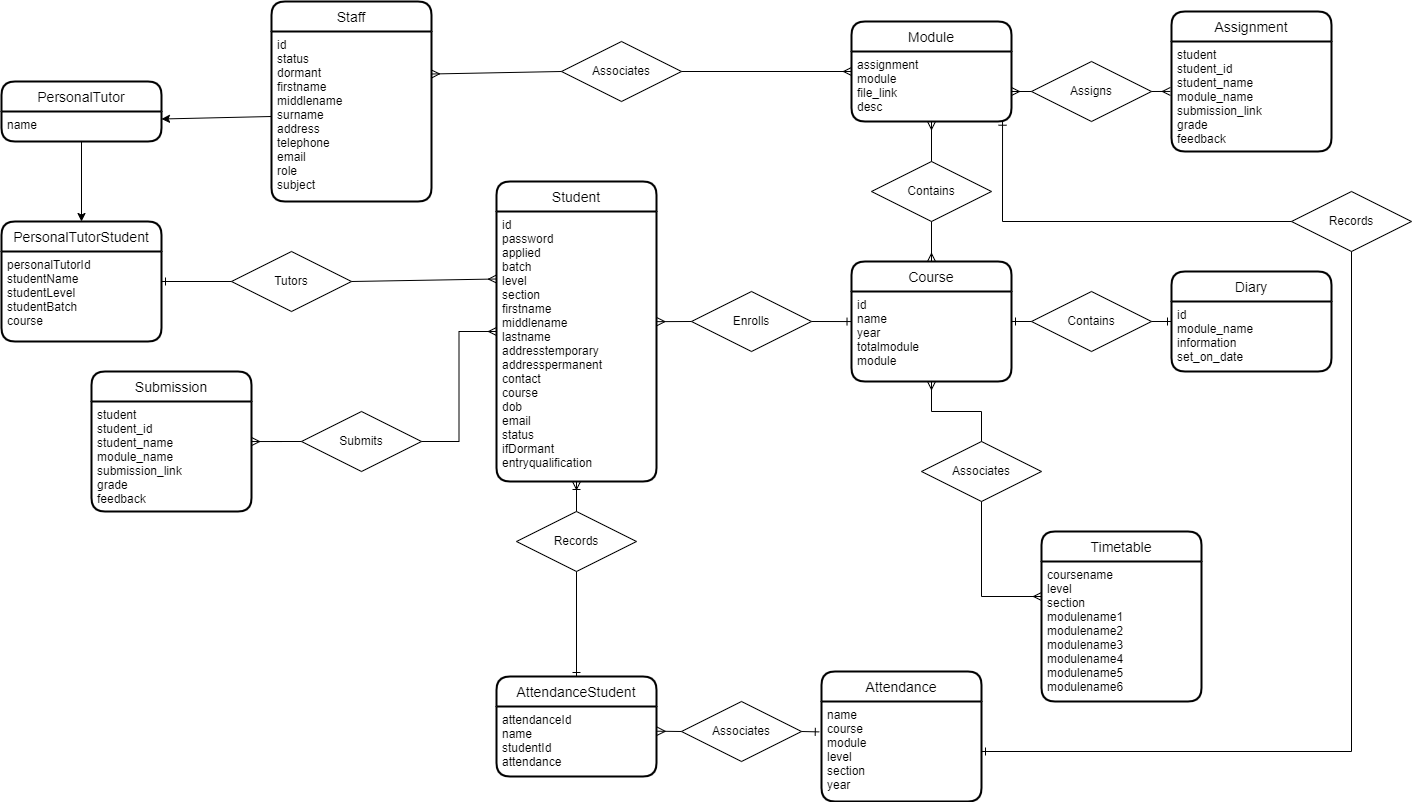


BON Dynamic Diagram

### System Database Design

* + - * 1. E-R Model

Here, we have the ER model for the database implemented in this system. As can be seen, admin is a table in this database has no relation with any other table.



E-R Model of the database

### Attribute Listings

Mongo DB

admin

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Datatype | Constraint | Default |
| admin\_id | NUMBER | UNIQUE |  |
| password | STRING | NOT NULL |  |

student

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Datatype | Constraint | Default |
| student\_id | NUMBER(8) | UNIQUE |  |
| firstname | STRING | NOT NULL |  |
| middlename | STRING |  |  |
| surname | STRING | NOT NULL |  |
| address | STRING | NOT NULL |  |
| contact | STRING | NOT NULL |  |
| join\_year | STRING | NOT NULL |  |
| section | STRING | NOT NULL |  |
| course\_id\* | NUMBER(8) | NOT NULL |  |
| email | STRING | NOT NULL |  |

staff

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Datatype | Constraint | Default |
| staff\_id | NUMBER | UNIQUE |  |
| firstname | STRING | NOT NULL |  |
| middlename | STRING | NOT NULL |  |
| surname | STRING | NOT NULL |  |
| address | STRING | NOT NULL |  |
| telephone | STRING | NOT NULL |  |
| email | STRING | NOT NULL |  |
| role | STRING | NOT NULL |  |
| subject | STRING | NOTNULL |  |
| status | STRING | NOTNULL |  |
| dormat | STRING | NOTNULL |  |

course

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Datatype | Constraint | Default |
| course\_id | NUMBER | UNIQUE |  |
| course\_name | STRING | NOT NULL |  |

module

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Datatype | Constraint | Default |
| module\_id | NUMBER | UNIQUE |  |
| module\_name | STRING | NOT NULL |  |
| staff | OBJECT | NOTNULL |  |
| link | STRING |  |  |

submission

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Datatype | Constraint | Default |
| Student\_id | NUMBER | NOT NULL |  |
| studentName | STRING | NOTNULL |  |
| studentLevel | STRING | NOTNULL |  |
| studentBatch | STRING | NOTNULL |  |
| course | STRING | NOTNULL |  |

assignment

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Attribute | | Datatype | Constraint | | Default | |
| assignment | | STRING | NOTNULL | |  | |
| module | | OBJECT | NOTNULL | |  | |
| file\_link | | STRING | NOTNULL | |  | |
| desc | STRING | | | NOT NULL | |  |

attendance

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Datatype | Constraint | Default |
| attendance\_id | STRING | UNIQUE |  |
| name | STRING | NOT NULL |  |
| studentId | STRING | NOT NULL |  |
| year | STRING | NOT NULL |  |
| batch | STRING | NOT NULL |  |

personalTutor

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Datatype | Constraint | Default |
| name | STRING | NOT NULL |  |

personalTutorStudent

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Datatype | Constraint | Default |
| personaltutor\_id | NUMBER | NOT NULL |  |
| studentname | STRING | NOTNULL |  |
| module\_name | STRING | NOTNULL |  |
| submission\_link | STRING | NOTNULL |  |
| grade | STRING | NOTNULL |  |
| feedback | STRING |  |  |

diary

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Datatype | Constraint | Default |
| id | NUMBER | UNIQUE |  |
| module\_name | STRING | NOT NULL |  |
| information | STRING | NOT NULL |  |
| set\_on\_date | STRING | NOT NULL |  |

timetable

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Datatype | Constraint | Default |
| coursename | NUMBER | UNIQUE |  |
| level | STRING | NOT NULL |  |
| section | STRING | NOT NULL |  |
| modulename1 | STRING | NOT NULL |  |
| modulename2 | STRING | NOT NULL |  |
| modulename3 | STRING | NOT NULL |  |
| modulename4 | STRING | NOT NULL |  |
| modulename5 | STRING | NOT NULL |  |
| modulename6 | STRING | NOT NULL |  |

attendanceStudent

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Datatype | Constraint | Default |
| attendanceId | NUMBER | UNIQUE |  |
| name | STRING | NOT NULL |  |
| studentId | STRING | NOT NULL |  |
| attendance | STRING | NOT NULL |  |

# 4. System Interface Design

## 4.1 Draft Interface Design (Record Management)

### 4.1.1 Wireframes

Below are the wireframes of the Record Management.

A picture containing text, whiteboard

Description automatically generated

Figure 1 Wireframes Login Page

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

I have included all the remaining wireframes in the Wireframes folder.

### 4.1.2 System Navigation Diagram (Record Management)

A screenshot of a computer

Description automatically generated with low confidence

Figure 43 System Navigation Diagram (Record Management)

### Level 0 DFD

Diagram

Description automatically generated

|  |  |  |
| --- | --- | --- |
| Diagram  Description automatically generated with medium confidence  Figure 44 Login Page of Record Management System | | |
| It contains the logo of the Woolands University College. | Below logo, it contains input text to enter user id. | |
| Below it contains input text to enter password. | | It contains login button to enter the webpage |

|  |  |  |
| --- | --- | --- |
| 4.1.3 System Screen Mock-ups (Record Management)1. This is the Nav bar enables to navigate to other section of the website. | | 2. This bold heading indicates that we are currently present in the list where only the enrolled student are present. |
| 3. This create button enables to create the new student profile. | | 4. This archived button enables user to see the archived list of student profile. |
| Figure 46 Enrolled Section of Student in Record Management System | | |
| 6. This gives the name of the student of the particular item. |
| 7. This course dropdown enables user to filter the student list of particular course from all profile. | | 8. This edit button makes user to edit the student profile. |
| 9. This archived button makes user to archive the particular profile . | | 10. This is the Log out button which enables user to get out of the system. |

|  |  |  |  |
| --- | --- | --- | --- |
| 1. Input area to add id of student | 1. Radio button to choose status fo student | | |
| 3. Radio button to choose reason of dormancy | 4. This requires the batch of the student. | | |
| Figure 48 Create Student data in Record Management System | | | |
| 5. This input area asks the level of the student. | 7. This input area ask the contact field from the user. | | |
| 6. Input area to add permanent address | 7. Input area to add contact of student. | | |
| 8. This input area asks the course of the student. | 9. Input area to add Date of birth. | | |
| 10. Input area to enter the email | | 11. This dropdown allows user to add entry qualifaction. | 13. This cancel button enables user to cancel the particular student profile from saving in the database. |
| 12. This create button enables user to save the student profile in the database. | |

|  |  |  |  |
| --- | --- | --- | --- |
| Figure 50 Staff Data in Record Management System | | | |
| 1. This gives the staff name | | 2. This button creates the staff record. | |
| 1. This button enables user to see the archived staff records | | 1. This search button enables the user to search among the staff records | |
| 5. This helps to assign the module to staff. | | | 6. This button edits the properties of the profile in the database. |
| 7. This button archives the that particular staff profile. | 8. This button deletes the staff profile. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 1. This allows the user to add the id of the staff. | 2. radio button to select the record status | | |
| 3. Radio button to select the reason for dormancy | 4. this allows the user to add the first name of the staff | | |
| Figure 52 Create staff Record in Record Management System | | | |
| 5. This allows the user to add the middle name of the staff. | 6. This allows the user to add the last name of the staff. | | |
| 7. Textarea to add address. | 8. This allows the user to enter the contact no of the staff. | | |
| 9. This allows the user to add the email of the staff | | 10. This allows the user to select the role for that particular staff. | 11. text area to add subject |
| 12. This create button creates the new staff data with filed properties in the database. | | 13. This cancel button cancel the staff data storing from the database. | |

Remaining mockups are available in inside the Mockups folder.

### 4.1.4 System Activity Event Diagram (Record Management)

|  |  |
| --- | --- |
| Diagram  Description automatically generated  Figure 86 System Event Diagram- Login System | Display 1 |
| Display 2 |
| Display 3  Graphical user interface, application  Description automatically generated |

|  |  |
| --- | --- |
| Assign Assignment Operation  Diagram  Description automatically generated  Figure 87 System Activity Event Diagram Assign Assignment | Display 1  Graphical user interface, application, Teams  Description automatically generated |

|  |  |
| --- | --- |
| Mark Assignment Operation  Diagram  Description automatically generated  Figure 88 System Activity Event Diagram Mark Assignment | Display 1  Graphical user interface, application  Description automatically generated |

|  |  |
| --- | --- |
| Create Student Profile Operation  Diagram  Description automatically generated  Figure 89 System Activity Event Diagram Create student profile | Display 1  Graphical user interface, application, Teams  Description automatically generated |

|  |  |
| --- | --- |
| Create staff profile Operation  Diagram  Description automatically generated  Figure 90 System Activity Event Diagram Create Staff Profile | Display 1  Graphical user interface, application, Teams  Description automatically generated |
| Create Course Structure Operation  Diagram  Description automatically generated  Figure 91 System Activity Event Diagram Create Course Structure | Display 1  Graphical user interface, application, Teams  Description automatically generated |

|  |  |
| --- | --- |
| Create Timetable Operation  Diagram  Description automatically generated  Figure 92 System Activity Event Diagram Create Timetable | Display 1  Table  Description automatically generated |
| Create Module Operation  Diagram  Description automatically generated  Figure 93 System Activity Event Diagram Create Module | Display 1  A picture containing shape  Description automatically generated |

|  |  |
| --- | --- |
| Create Attendance Operation  Diagram  Description automatically generated  Figure 94 System Activity Event Diagram Create Attendance | Display 1  Graphical user interface, application, Teams  Description automatically generated |

|  |  |
| --- | --- |
| Create Diary Operation  Diagram  Description automatically generated  Figure 95 System Activity Event Diagram Create Diary | Display 1  Graphical user interface, application, Teams  Description automatically generated |

|  |  |
| --- | --- |
| Generate Report Operation  Diagram  Description automatically generated  Figure 96 System Activity Event Diagram Mark Assignment | Display 1  Diagram  Description automatically generated with low confidence |

### 4.2 Design Revisions (Record Management)

The design above presented are the belonging of the Record Management.

## 4.4 Draft Interface Design (Student Record/Information Portal)

### 4.4.1 Wireframes

Below are the wireframes of the Student Record/Information Portal.

A piece of paper with writing on it

Description automatically generated

Figure 97 Wireframe of Courses Section (Student Portal)

A close-up of a book

Description automatically generated with low confidence

Figure 98 Wireframe (Student Portal) Report Section

A piece of paper with writing on it

Description automatically generated

Figure 99 Wireframe (Student Portal) of Module Section

A piece of paper with writing on it

Description automatically generated with medium confidence

Figure 100 Wireframe (Student Portal) of Diary Section

A piece of paper with writing on it

Description automatically generated

Figure 101 Wireframe (Student Portal) of Assignment Section

### 4.4.2 System Navigation Diagram (Student Record/Information Portal)

Chart, diagram, box and whisker chart

Description automatically generated

Figure 106 Wireframe (Student Portal) Student Portal

### 4.4.3 System Screen Mock-ups (Student Record/Information Portal)

|  |  |
| --- | --- |
| 1. This is the Nav bar enables to navigate to other section of the website. | 2. This is the Log out button which enables user to get out of the system. |
| 3. This displays the name of the course. | 4. This displays the title of the subsection. |
| Graphical user interface, application  Description automatically generated  Figure 107 Courses section in Student Portal | |
| 5. This button helps user to applying the particular course. | |

|  |  |
| --- | --- |
| 1. This is the Nav bar enables to navigate to other section of the website. | 2. This is the Log out button which enables user to get out of the system. |
| 3. This displays the module name. | 4. This displays the title of the current subsection. |
| Graphical user interface, application  Description automatically generated  Figure 108 Module Section in Student Portal | |
| 5. This button enables the user to view the module material. | |

|  |  |
| --- | --- |
| 1. This is the Nav bar enables to navigate to other section of the website. | 2. This is the Log out button which enables user to get out of the system. |
| 3. This displays the assignment name. | 4. This displays the status of the submission enables to submit or view their assignment. |
| Graphical user interface, application  Description automatically generated  Figure 109 Assignment section in Student Portal | |
| 5. This displays the title of the current subsection. | |

|  |  |
| --- | --- |
| 1. This is the Nav bar enables to navigate to other section of the website. | 2. This is the Log out button which enables user to get out of the system. |
| 3. This displays title of the current subsecction. | 4. This displays the particular module name. |
| Graphical user interface, application  Description automatically generated  Figure 110 Module section in student portal | |
| 5. This area gives the google drive link of the module material. | 6. This okay button navigates back to the previous page. |

### 4.4.4 System Activity Event Diagram (Student Record/Information Portal)

|  |  |
| --- | --- |
| Submit Assignment Operation  Diagram  Description automatically generated  Figure 116 System Activity Event Diagram Submit Assignment Operation | Display 1  Graphical user interface, text, application  Description automatically generated |

### 4.5 Design Revisions (Student Record/Information Portal)

The design above presented are the belonging of the Record Management

# 5. System Build and Technical Notes

We have chosen our group work as a MERN project. Both Record Management system and student portal are made on MERN Stack. M for Mongo Db which is very popular for NoSQL which stores the records in the format of documents. We use mongoose to interact with Mongo Db. E for Express, known as Node JS framework for backend and acts as middleware. R stands for React JS, popular for JS Frontend library backed by Facebook and N for Node JS, which provide runtime environment in server for JS code. In our project, frontend was from react where we have used react hooks which replaced workloads of writing lifecycle methods with classes. All division are replaced by the components. React routing is also used here for navigation. In backend, NPM allowed us to run our project in our pc. Express made our code simpler by using routes which gave path for communicating with database, controller which handled CRUD, models, helped for creating the schema.

## 5.1 Technically Difficult Code Section

As this was our first project in the MERN and with the low time frame, we must skip very complex operation from our project. But there is a code in attendance module where we have implemented the importing of the documents from other collections and performing the crud operations in it.

### 5.1.1 Create Attendance Record

A screenshot of a computer

Description automatically generated with medium confidence

Here, after creating the attendance record the data is send to the server. At server, it checks the data from the following code and if it is fine and perform the above operation. This was very complex for us as it involves finding the documents from the other collection and saving it in own section.

Above it finds the students record which has same course, level, batch, section as data send from the client while creating attendance record. When it gets the data from student collection, it will save the documents to the attendance student collection.

### 5.1.2 Take attendance

A screenshot of a computer

Description automatically generated with medium confidence

Though it is not hard as previous, but it was something where we were performing operations on the array which was very new for us. Here we get the today’s date and check the date status of the current date. If it had taken attendance earlier, then it will toggle the value otherwise it will push the date and status to the array.

Final Screen image are present in the Mockup folder

# 6. Test Strategy

## 6.1 Overview of Test Strategy

### Black Box Testing:

This is popular method of testing which contemplates about the external appearance and functionalities of the system. Methodology and inner functionalities are not kept in consideration. Name suggest that inside the box can’t be seen because of black box. So, this ability makes it capable enough to find the bugs in data structures and database interaction. This test is done on the scope of customer requirements.

### White Box Testing:

White box testing gives way to interact the product or project with testers for inspection and verification of the inner interaction between the components with external environment. Internal design and structure are taken in consideration. Bugs in I/O are solved easily using this testing. Testers and developer prefer this testing as it plays important role to improve the usability and security.

The below test result of the record management system and student portal was done in desktop browser, which showed everything was as intended. These tests were manually performed to check each functionality.

## Test Results for Record Management System:

### Test Result for Attendance:

|  |  |  |
| --- | --- | --- |
| SN | Requirements | Test Scenario |
| 1 | Build Attendance Module for Record Management System | Check create function |
| 2 | Check Take Attendance Function |
| 3 | Check Delete Function |
| 4 | Check Edit option |
| 5 |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SN# | ID | Test Case Name | Pre-Requisites | Steps | Activity | Expected Result | Actual Results | Priority | Author | Status | Remarks |
| 1 | 1\_01 | Check create function with valid student information | Valid Course, Module, Level, Section and Batch data of students. | Step 1 | Click on the create button | Redirects to create page with form. | Page was redirected to allow to create attendance | P1 | Niraula, Rohan | Pass |  |
|  |  |  |  | Step 2 | Fill up the form with valid information. And click on submit | Redirects to previous page. | Redirections was there to previous page. | P1 | Niraula, Rohan | Pass |  |
|  |  |  |  | Step 3 | Click on the Take attendance button | Redirects to the page with list of the student | Redirects to the page with list of the student | P1 | Niraula, Rohan | Pass |  |
| 1 | 1\_02 | Check create function with invalid student information | Invalid Course, Module, Level, Section and Batch data of students. | Step 1 | Click on the create button | Redirects to create page with form. | Page was redirected to allow to create attendance | P1 | Niraula, Rohan | Pass |  |
|  |  |  |  | Step 2 | Fill up the form with valid information. And click on submit | Redirects to previous page. | Redirections was there to previous page. | P1 | Niraula, Rohan | Pass |  |
|  |  |  |  | Step 3 | Click on the Take attendance button | Redirects to the page with no list of the students. | Redirects to the page with no list of the students | P1 | Niraula, Rohan | Pass |  |
| 2 | 2\_01 | Check Take Attendance Function | Valid Attendance record with list of students | Step 1 | Click on the Take Attendance Button | Redirects to the page with list of the student | Redirects to the page with list of the student | P1 | Niraula, Rohan | Pass |  |
|  |  |  |  | Step 2 | Click on the Absent Button of any student | Button text changes to Present | Button text changes to Present | P1 | Niraula, Rohan | Pass |  |
|  |  |  |  | Step 3 | Again click on the present Button of that student | Button text changes to Absent | Button text changes to Absent | P1 | Niraula, Rohan | Pass |  |
| 3 | 3\_01 | Check delete function of attendance | Valid Attendance record | Step 1 | Click on the text of the attendance sheet | Redirects to the page showing the attendance details | Redirects to the page showing the attendance details | P1 | Niraula, Rohan | Pass |  |
|  |  |  |  | Step 2 | Click on the delete Record option | Takes to the previous page with removal of that attendance record | Takes to the previous page with removal of that attendance record | P1 | Niraula, Rohan | Pass |  |
| 4 | 4\_01 | Check Edit function of attendance | Valid attendance record | Step 1 | Click on the edit button on attendance record | Redirects to create page with form filled with its details | Redirects to create page with form filled with its details | P1 | Niraula, Rohan | Pass |  |
|  |  |  |  | Step 2 | Update on any field and click submit button | Redirects to the previous page | Redirects to the previous page | P1 | Niraula, Rohan | Pass |  |
|  |  |  |  | Step 3 | Click on that record to see the changes | Changes are seen | Changes are seen | P1 | Niraula, Rohan | Pass |  |

### Test Result for Student:

|  |  |  |
| --- | --- | --- |
| SN | Requirements | Test Scenario |
| 1 | Build Student Module for Record Management System | Check create function |
| 2 | Check Delete Function |
| 3 | Check Edit option |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SN# | ID | Test Case Name | Pre-Requisites | Steps | Activity | Expected Result | Actual Results | Priority | Author | Status | Remarks |
| 1 | 1\_01 | Check create function | Must be logged in and be present in student dashboard  Valid student information data | Step 1 | Click on the create button on student dashboard | Redirects to create page with form. | Page was redirected to allow to create student record | P1 | Niraula, Rohan | Pass |  |
|  |  |  |  | Step 2 | Fill up the form with valid information. And click on submit | Redirects to previous page with showing that student record in the list. | Redirections was there to previous page with showing that student record in the list. | P1 | Niraula, Rohan | Pass |  |
| 2 | 2\_01 | Check delete function of student | Valid Student record | Step 1 | Click on any student record | Redirects to the page showing the student details | Redirects to the page showing the student details | P1 | Niraula, Rohan | Pass |  |
|  |  |  |  | Step 2 | Click on the delete Record option | Takes to the previous page with removal of that student record | Takes to the previous page with removal of that student record | P1 | Niraula, Rohan | Pass |  |
| 3 | 3\_01 | Check Edit function of student | Valid student record | Step 1 | Click on the edit button on student record | Redirects to create page with form filled with its details | Redirects to create page with form filled with its details except dormant and status | P1 | Niraula, Rohan | Fail | Need for updation |
|  |  |  |  | Step 2 | Update on any field and click submit button | Redirects to the previous page | Redirects to the previous page | P1 | Niraula, Rohan | Pass |  |
|  |  |  |  | Step 3 | Click on that record to see the changes | Changes are seen | Changes are seen | P1 | Niraula, Rohan | Pass |  |

### Test Case for Personal Tutor:

|  |  |  |
| --- | --- | --- |
| SN | Requirements | Test Scenario |
| 1 | Build Personal Tutor Module for Record Management System | Check create function |
| 2 | Check Assign Function |
| 3 | Check Delete Function |
| 4 | Check Edit option |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SN# | ID | Test Case Name | Pre-Requisites | Steps | Activity | Expected Result | Actual Results | Priority | Author | Status | Remarks |
| 1 | 1\_01 | Check create function with personal tutor | Valid personal tutor name. | Step 1 | Click on the create button | Redirects to create page with form. | Page was redirected to allow to create attendance | P1 | Niraula, Rohan | Pass |  |
|  |  |  |  | Step 2 | Fill up the form with valid information. And click on submit | Redirects to previous page. | Redirections was there to previous page. | P1 | Niraula, Rohan | Pass |  |
| 2 | 2\_01 | Check Assign Function | Valid personal tutor record | Step 1 | Click on the Take Attendance Button | Redirects to the page with list of the student if created earlier | Redirects to the page with list of the student if created earlier | P1 | Niraula, Rohan | Pass |  |
|  |  |  |  | Step 2 | Click on the create option at that page | Redirects to page to create a student data with forms | Redirects to page to create a student data with forms | P1 | Niraula, Rohan | Pass |  |
|  |  |  |  | Step 3 | Fill up the form with valid information. And click on submit | Redirects to the Step 1 | Redirects to the Step 1 | P1 | Niraula, Rohan | Pass |  |
|  |  |  |  | Step 4 | Click on that newly created personal tutor | Newly created student record is present there | Newly created student record is present there | P1 | Niraula, Rohan | Pass |  |
| 3 | 3\_01 | Check delete function of personal tutor record | Valid personal tutor record | Step 1 | Click on the text of the personal tutor record | Redirects to the page showing the Personal Tutor details | Redirects to the page showing the Personal Tutor details | P1 | Niraula, Rohan | Pass |  |
|  |  |  |  | Step 2 | Click on the delete Record option | Takes to the previous page with removal of that Personal Tutor record | Takes to the previous page with removal of that Personal Tutor record | P1 | Niraula, Rohan | Pass |  |
| 4 | 4\_01 | Check Edit function of Personal Tutor | Valid Personal Tutor record | Step 1 | Click on the edit button on attendance record | Redirects to create page with form filled with its details | Redirects to create page with form filled with its details | P1 | Niraula, Rohan | Pass |  |
|  |  |  |  | Step 2 | Update on any field and click submit button | Redirects to the previous page | Redirects to the previous page | P1 | Niraula, Rohan | Pass |  |
|  |  |  |  | Step 3 | Click on that record to see the changes | Changes are seen | Changes are seen | P1 | Niraula, Rohan | Pass |  |

### Test Case for Staff:

|  |  |  |
| --- | --- | --- |
| SN | Requirements | Test Scenario |
| 1 | Build Staff Module for Record Management System | Check create function |
| 2 | Check Delete Function |
| 3 | Check Edit option |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SN# | ID | Test Case Name | Pre-Requisites | Steps | Activity | Expected Result | Actual Results | Priority | Author | Status | Remarks |
| 1 | 1\_01 | Check create function with staff | Staff details | Step 1 | Click on the create button | Redirects to create page with form. | Page was redirected to allow to create staff | P1 | Niraula, Rohan | Pass |  |
|  |  |  |  | Step 2 | Fill up the form with staff information. And click on submit | Redirects to previous page. | Redirections was there to previous page. | P1 | Niraula, Rohan | Pass |  |
| 2 | 2\_01 | Check delete function of staff record | staff record | Step 1 | Click on the text of the staff record | Redirects to the page showing the Staff details | Redirects to the page showing the Staff details | P1 | Niraula, Rohan | Pass |  |
|  |  |  |  | Step 2 | Click on the delete Record option | Takes to the previous page with removal of that Staff record | Takes to the previous page with removal of that Staff record | P1 | Niraula, Rohan | Pass |  |
| 3 | 3\_01 | Check Edit function of Staff | Staff record | Step 1 | Click on the edit button on staff record | Redirects to create page with form filled with its details | Redirects to create page with form filled with its details | P1 | Niraula, Rohan | Pass |  |
|  |  |  |  | Step 2 | Update on any field and click submit button | Redirects to the previous page | Redirects to the previous page | P1 | Niraula, Rohan | Pass |  |
|  |  |  |  | Step 3 | Click on that record to see the changes | Changes are seen | Changes are seen | P1 | Niraula, Rohan | Pass |  |

Test Case for Course:

|  |  |  |
| --- | --- | --- |
| SN | Requirements | Test Scenario |
| 1 | Build Course Module for Record Management System | Check create function |
| 2 | Check Delete Function |
| 3 | Check Edit option |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SN# | ID | Test Case Name | Pre-Requisites | Steps | Activity | Expected Result | Actual Results | Priority | Author | Status | Remarks |
| 1 | 1\_01 | Check create function with course | Course Details | Step 1 | Click on the create button | Redirects to create page with form. | Page was redirected to allow to create course | P1 | Niraula, Rohan | Pass |  |
|  |  |  |  | Step 2 | Fill up the form with course information. And click on submit | Redirects to previous page. | Redirections was there to previous page. | P1 | Niraula, Rohan | Pass |  |
| 2 | 2\_01 | Check delete function of course record | course record | Step 1 | Click on the text of the course record | Redirects to the page showing the Course details | Redirects to the page showing the Course details | P1 | Niraula, Rohan | Pass |  |
|  |  |  |  | Step 2 | Click on the delete Record option | Takes to the previous page with removal of that Course record | Takes to the previous page with removal of that Course record | P1 | Niraula, Rohan | Pass |  |
| 3 | 3\_01 | Check Edit function of Course | Course record | Step 1 | Click on the edit button on course record | Redirects to create page with form filled with its details | Redirects to create page with form filled with its details | P1 | Niraula, Rohan | Pass |  |
|  |  |  |  | Step 2 | Update on any field and click submit button | Redirects to the previous page | Redirects to the previous page | P1 | Niraula, Rohan | Pass |  |
|  |  |  |  | Step 3 | Click on that record to see the changes | Changes are seen | Changes are seen | P1 | Niraula, Rohan | Pass |  |

### Test Case for Module:

|  |  |  |
| --- | --- | --- |
| SN | Requirements | Test Scenario |
| 1 | Build Module for Record Management System | Check create function |
| 2 | Check Delete Function |
| 3 | Check Edit option |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SN# | ID | Test Case Name | Pre-Requisites | Steps | Activity | Expected Result | Actual Results | Priority | Author | Status | Remarks |
| 1 | 1\_01 | Check create function with module | Module Details | Step 1 | Click on the create button | Redirects to create page with form. | Page was redirected to allow to create module | P1 | Niraula, Rohan | Pass |  |
|  |  |  |  | Step 2 | Fill up the form with module information. And click on submit | Redirects to previous page. | Redirections was there to previous page. | P1 | Niraula, Rohan | Pass |  |
| 2 | 2\_01 | Check delete function of module record | module record | Step 1 | Click on the text of the module record | Redirects to the page showing the Module details | Redirects to the page showing the Module details | P1 | Niraula, Rohan | Pass |  |
|  |  |  |  | Step 2 | Click on the delete Record option | Takes to the previous page with removal of that Module record | Takes to the previous page with removal of that Module record | P1 | Niraula, Rohan | Pass |  |
| 3 | 3\_01 | Check Edit function of Module | Module record | Step 1 | Click on the edit button on module record | Redirects to create page with form filled with its details | Redirects to create page with form filled with its details | P1 | Niraula, Rohan | Pass |  |
|  |  |  |  | Step 2 | Update on any field and click submit button | Redirects to the previous page | Redirects to the previous page | P1 | Niraula, Rohan | Pass |  |
|  |  |  |  | Step 3 | Click on that record to see the changes | Changes are seen | Changes are seen | P1 | Niraula, Rohan | Pass |  |

### Test Case for Assignment:

|  |  |  |
| --- | --- | --- |
| SN | Requirements | Test Scenario |
| 1 | Build Assignment for Record Management System | Check create function |
| 2 | Check Delete Function |
| 3 | Check Edit option |
| 4 | Check Mark option |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SN# | ID | Test Case Name | Pre-Requisites | Steps | Activity | Expected Result | Actual Results | Priority | Author | Status | Remarks |
| 1 | 1\_01 | Check create function with assignment | Assignment details | Step 1 | Click on the create button | Redirects to create page with form. | Page was redirected to allow to create assignment | P1 | Niraula, Rohan | Pass |  |
|  |  |  |  | Step 2 | Fill up the form with assignment information. And click on submit | Redirects to previous page. | Redirections was there to previous page. | P1 | Niraula, Rohan | Pass |  |
| 2 | 2\_01 | Check delete function of assignment record | Assignment record | Step 1 | Click on the text of the assignment record | Redirects to the page showing the Assignment details | Redirects to the page showing the Assignment details | P1 | Niraula, Rohan | Pass |  |
|  |  |  |  | Step 2 | Click on the delete Record option | Takes to the previous page with removal of that Assignment record | Takes to the previous page with removal of that Assignment record | P1 | Niraula, Rohan | Pass |  |
| 3 | 3\_01 | Check Edit function of Assignment | Assignment record | Step 1 | Click on the edit button on assignment record | Redirects to create page with form filled with its details | Redirects to create page with form filled with its details | P1 | Niraula, Rohan | Pass |  |
|  |  |  |  | Step 2 | Update on any field and click submit button | Redirects to the previous page | Redirects to the previous page | P1 | Niraula, Rohan | Pass |  |
|  |  |  |  | Step 3 | Click on that record to see the changes | Changes are seen | Changes are seen | P1 | Niraula, Rohan | Pass |  |
| 4 | 4\_01 | Check mark assignment function | Created assignment and student has submitted it | Step 1 | Click on the mark button on the assignment record | Redirects to the page where list of submitted assignment are shown | Redirects to the page where list of submitted assignment are shown | P1 | Niraula, Rohan | Pass |  |
|  |  |  |  | Step 2 | Click on the mark button on the student submitted assignment record | Redirects to the page where google link of the assignment is shown | Redirects to the page where google link of the assignment is shown | P1 | Niraula, Rohan | Pass |  |
|  |  |  |  | Step 3 | Mark the assignment with the grade field. And click submit button. | Page redirects to the previous page | Page redirects to the previous page | P1 | Niraula, Rohan | Pass |  |

### Test Case for Timetable:

|  |  |  |
| --- | --- | --- |
| SN | Requirements | Test Scenario |
| 1 | Build Timetable for Record Management System | Check create function |
| 2 | Check Delete Function |
| 3 | Check Edit option |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SN# | ID | Test Case Name | Pre-Requisites | Steps | Activity | Expected Result | Actual Results | Priority | Author | Status | Remarks |
| 1 | 1\_01 | Check create function with timetable | Timetable Credential | Step 1 | Click on the create button | Redirects to create page with form. | Page was redirected to allow to create timetable | P1 | Niraula, Rohan | Pass |  |
|  |  |  |  | Step 2 | Fill up the form with timetable information. And click on submit | Redirects to previous page. | Redirections was there to previous page. | P1 | Niraula, Rohan | Pass |  |
| 2 | 2\_01 | Check deletes function of timetable record | timetable record | Step 1 | Click on the text of the timetable record | Redirects to the page showing the Timetable details | Redirects to the page showing the Timetable details | P1 | Niraula, Rohan | Pass |  |
|  |  |  |  | Step 2 | Click on the delete Record option | Takes to the previous page with removal of that Timetable record | Takes to the previous page with removal of that Timetable record | P1 | Niraula, Rohan | Pass |  |
| 3 | 3\_01 | Check Edit function of Timetable | Timetable record | Step 1 | Click on the edit button on timetable record | Redirects to create page with form filled with its details | Redirects to create page with form filled with its details | P1 | Niraula, Rohan | Pass |  |
|  |  |  |  | Step 2 | Update on any field and click submit button | Redirects to the previous page | Redirects to the previous page | P1 | Niraula, Rohan | Pass |  |
|  |  |  |  | Step 3 | Click on that record to see the changes | Changes are seen | Changes are seen | P1 | Niraula, Rohan | Pass |  |

### Test Case for Diaries:

|  |  |  |
| --- | --- | --- |
| SN | Requirements | Test Scenario |
| 1 | Build diaries for Record Management System | Check create function |
| 2 | Check Delete Function |
| 3 | Check Edit option |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SN# | ID | Test Case Name | Pre-Requisites | Steps | Activity | Expected Result | Actual Results | Priority | Author | Status | Remarks |
| 1 | 1\_01 | Check create function with diary | Diary Credential | Step 1 | Click on the create button | Redirects to create page with form. | Page was redirected to allow to create diary | P1 | Niraula, Rohan | Pass |  |
|  |  |  |  | Step 2 | Fill up the form with diary information. And click on submit | Redirects to previous page. | Redirections was there to previous page. | P1 | Niraula, Rohan | Pass |  |
| 2 | 2\_01 | Check deletes function of diary record | diary record | Step 1 | Click on the text of the diary record | Redirects to the page showing the Diary details | Redirects to the page showing the Diary details | P1 | Niraula, Rohan | Pass |  |
|  |  |  |  | Step 2 | Click on the delete Record option | Takes to the previous page with removal of that Diary record | Takes to the previous page with removal of that Diary record | P1 | Niraula, Rohan | Pass |  |
| 3 | 3\_01 | Check Edit function of Diary | Diary record | Step 1 | Click on the edit button on diary record | Redirects to create page with form filled with its details | Redirects to create page with form filled with its details | P1 | Niraula, Rohan | Pass |  |
|  |  |  |  | Step 2 | Update on any field and click submit button | Redirects to the previous page | Redirects to the previous page | P1 | Niraula, Rohan | Pass |  |
|  |  |  |  | Step 3 | Click on that record to see the changes | Changes are seen | Changes are seen | P1 | Niraula, Rohan | Pass |  |

### Test Case for Login:

|  |  |  |
| --- | --- | --- |
| SN | Requirements | Test Scenario |
| 1 | Build Login for Record Management System | Check login feature |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SN# | ID | Test Case Name | Pre-Requisites | Steps | Activity | Expected Result | Actual Results | Priority | Author | Status | Remarks |
| 1 | 1\_01 | Check Login feature with valid credentials | Valid credentials: user id and password | Step 1 | Type user id and password in the form and click login. | Enters to the application in student page | Enters to the application in student page | P1 | Niraula, Rohan | Pass |  |
| 1 | 1\_02 | Check Login feature with invalid credentials | Invalid credentials (username and password) | Step 1 | Type user id and password in the form and click login. | Alert message with “Credentials don’t match” will appear | Alert message with “Credentials don’t match” will appear | P1 | Niraula, Rohan | Pass |  |

### Test Case for Report Section:

|  |  |  |
| --- | --- | --- |
| SN | Requirements | Test Scenario |
| 1 | Build Report Module for Record Management System | Check generate report function |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SN# | ID | Test Case Name | Pre-Requisites | Steps | Activity | Expected Result | Actual Results | Priority | Author | Status | Remarks |
| 1 | 1\_01 | Check generate report function for student | Credentials of Student | Step 1 | Write the credentials of the student in the form and click generate | Generate a pdf report of student | Generate a pdf report of student | P1 | Niraula, Rohan | Pass |  |
|  |  |  |  | Step 2 | Open the file and check whether it belongs to that student or not | Belongs to the student whose credentials was entered | Belongs to the student whose credentials was entered | P1 | Niraula, Rohan | Pass |  |
|  | 1\_02 | Check generate report function for staff | Credentials of Staff | Step 1 | Write the credentials of the staff in the form and click generate | Generate a pdf report of staff | Generate a pdf report of staff | P1 | Niraula, Rohan | Pass |  |
|  |  |  |  | Step 2 | Open the file and check whether it belongs to that staff or not | Belongs to the staff whose credentials was entered | Belongs to the staff whose credentials was entered | P1 | Niraula, Rohan | Pass |  |

## Test Cases for Student Portal:

### Test Case for Courses:

|  |  |  |
| --- | --- | --- |
| SN | Requirements | Test Scenario |
| 1 | Build course module for Student Portal | Check enrolled course |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SN# | ID | Test Case Name | Pre-Requisites | Steps | Activity | Expected Result | Actual Results | Priority | Author | Status | Remarks |
| 1 | 1\_01 | Check enrolled course | Must be logged in student portal and in courses section | Step 1 | Check there enrolled button in course record | Button with text enrolled must be in the course record where the student was enrolled. | Button with text enrolled must be in the course record where the student was enrolled. | P1 | Niraula, Rohan | Pass |  |

### Test Case for Modules:

|  |  |  |
| --- | --- | --- |
| SN | Requirements | Test Scenario |
| 1 | Build Modules module for Student Portal | Check Module Material |
| 2 |  |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SN# | ID | Test Case Name | Pre-Requisites | Steps | Activity | Expected Result | Actual Results | Priority | Author | Status | Remarks |
| 1 | 1\_01 | Check Module Material | Must be logged in student portal and in modules section | Step 1 | Check there if there is any module record or not | There must be modules list | There was list of module records | P1 | Niraula, Rohan | Pass |  |
|  |  |  |  | Step 2 | Click on the view button of the module | Redirects to the next page and presents the module material | Redirects to the next page and presents the module material | P1 | Niraula, Rohan | Pass |  |

### Test Case for Assignment:

|  |  |  |
| --- | --- | --- |
| SN | Requirements | Test Scenario |
| 1 | Build Assignment module for Student Portal | View Assignment detail |
| 2 | Submit Assigment |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SN# | ID | Test Case Name | Pre-Requisites | Steps | Activity | Expected Result | Actual Results | Priority | Author | Status | Remarks |
| 1 | 1\_01 | Check Assignments detail if there is assignment issued by teacher | Must be logged in student portal and assignment section | Step 1 | Click on the assignment to see the assignment details if present | Page redirects and assignment detail will appear | Page was redirected and assignment detail was shown | P1 | Niraula, Rohan | Pass |  |
|  | 1\_02 | Check Assignments detail if there is no assignment issued by teacher | Must be logged in student portal and assignment section | Step 1 | Check there is assignment or not | No assignment will appear | No assignment was there | P1 | Niraula, Rohan | Pass |  |

### Test Case for Personal Tutor:

|  |  |  |
| --- | --- | --- |
| SN | Requirements | Test Scenario |
| 1 | Build Personal Tutor module for Student Portal | View Personal Tutor |
| 2 |  |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SN# | ID | Test Case Name | Pre-Requisites | Steps | Activity | Expected Result | Actual Results | Priority | Author | Status | Remarks |
| 1 | 1\_01 | View Personal Tutor if assigned | Must be logged in student portal and personal tutor section | Step 1 | Check if there is personal tutor | Personal tutor will be shown there | Personal tutor was shown | P1 | Niraula, Rohan | Pass |  |
|  | 1\_02 | View personal tutor if not assigned | Must be logged in student portal and assignment section | Step 1 | Check if there is personal tutor or not | No personal tutor will appear | No personal tutor was there | P1 | Niraula, Rohan | Pass |  |

### Test Case for Timetable:

|  |  |  |
| --- | --- | --- |
| SN | Requirements | Test Scenario |
| 1 | Build Timetable module for Student Portal | View Timetable |
|  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SN# | ID | Test Case Name | Pre-Requisites | Steps | Activity | Expected Result | Actual Results | Priority | Author | Status | Remarks |
| 1 | 1\_01 | View timetable if assigned | Must be logged in student portal and timetable section | Step 1 | Check if there is timetable | Timetable will be shown there | Timetable was shown | P1 | Niraula, Rohan | Pass |  |
|  | 1\_02 | View timetable if not assigned | Must be logged in student portal and timetable section | Step 1 | Check if there is timetable or not | No timetable will appear | No timetable was there | P1 | Niraula, Rohan | Pass |  |

Test Case for Diary:

|  |  |  |
| --- | --- | --- |
| SN | Requirements | Test Scenario |
| 1 | Build Diary module for Student Portal | View Diary |
|  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SN# | ID | Test Case Name | Pre-Requisites | Steps | Activity | Expected Result | Actual Results | Priority | Author | Status | Remarks |
| 1 | 1\_01 | View diary if assigned | Must be logged in student portal and diary section | Step 1 | Check if there is diary | Diary will be shown there | Diary was shown | P1 | Niraula, Rohan | Pass |  |
|  | 1\_02 | View diary if not assigned | Must be logged in student portal and diary section | Step 1 | Check if there is diary or not | No diary will appear | No diary was there | P1 | Niraula, Rohan | Pass |  |

### Test Case for Report:

|  |  |  |
| --- | --- | --- |
| SN | Requirements | Test Scenario |
| 1 | Build Report module for Student Portal | View Report |
|  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SN# | ID | Test Case Name | Pre-Requisites | Steps | Activity | Expected Result | Actual Results | Priority | Author | Status | Remarks |
| 1 | 1\_01 | View report if assigned | Must be logged in student portal and report section | Step 1 | Check if there is report | Report will be shown there | Report was shown | P1 | Niraula, Rohan | Pass |  |
|  | 1\_02 | View report if not assigned | Must be logged in student portal and report section | Step 1 | Check if there is report or not | No report will appear | No report was there | P1 | Niraula, Rohan | Pass |  |

# System evaluation Strategy

## 7.1 Usability Evaluation Strategy

### 7.1.1 Target Audience Selection/ Sample Size Selection

As you completed our designing part of app, be prepared strategy for its usability evaluation. For that react certain group of people to have samples of our app, so that we can know their opinion and evaluate usability of app accordingly.

### 7.1.2 Trial active task list creation\ design.

We asked those selected audience to use the app and to test the functions available there as a task for them. The audience were also requested to observe design part and provide feedback accordingly.

## 7.2 Initial system pilot trial/ results

### 7.2.1 Thinking aloud (observation and results)

From the trial of usability evaluation, we observed that the audience were satisfied with the sample use of our app. They were happy with the ease of using the app. We could observe the audience were quite impressed with the functionality of app and even with the speed of app. As a result we found that our project made audience happy and satisfied.

### 7.2.2 Post Event Walkthrough (Questions) and Results

Some questions that were asked during the trial of app are:

1) How do you feel about system design of project?

2) Was it difficult for you to use the system?

3) Do you find functionalities provided in the system helpful?

4) Anything that needs to be updated in system?

5) Were there any notable errors in system?

==Answer from one of our trial audience.

1) How do you feel about system design of project?

🡪The current design of system is nice and clean. It is able to match the competition in market. Functionalities available are also quite easy to locate.

2) Was it difficult for you to use the system?

🡪No, not at all, system is quite smooth and by the way nobody might feel difficult to use this.

3) Do you find functionalities provided in the system helpful?

🡪Yes, all available functions are quite helpful for course management system.

4) Anything that needs to be updated in system?

🡪The system is fine for now and I do not see any needed change for now. But with the increasing competition this might need to be updated in future.

5) Were there any notable errors in system?

🡪No, not any notable error is found during trial

## 7.3 Final System Trial/Results

### 7.3.1 Usability Questionnaire Design/ 7.3.2 Final Questionnaire (Used in Trials)

The evaluation includes some set of questionnaire to be asked to the stake holders. The questions that were asked during evaluation are:

🡪Have the system met your expectation?

🡪How easy was it to use the system?

🡪Is it easy to locate the functionalities?

🡪How good was the speed of system?

🡪Is it easy to locate the functionalities?

🡪Your final opinion on the system?

### 7.3.2 Questionnaire Results – Implications of Results

🡪These are the questionnaire prepared for evaluation

1. Have the system met your expectation?

* **☐ Yes**

**☐ No**

1. How easy was it to use the system rate from 1 to 5 (5 being the greatest)?

* **☐ 1**

**☐ 2**

**☐ 3**

**☐ 4**

**☐ 5**

3 .Is it easy to locate the functionalities?

* **☐ Yes**

**☐ No**

1. How good was the speed of system rate from 1 to 5 (5 being the greatest)?

* **☐ 1**

**☐ 2**

**☐ 3**

**☐ 4**

**☐ 5**

1. Is it easy to locate the functionalities?

* **☐ Yes**

**☐ No**

1. Your final opinion on the system?

* **☐ Poor**

**☐Satisfactory**

**☐Good**

**☐Very Good**

**☐Excellent**

# 8 Project Conclusion

With just due time of one month for the project we have completed our web application of students portal and course management system for Woodland university college.

This project was followed with the documentation in various topics that is required specifications, system design and analysis which full stack coding for colleges course management system with at least 10 models.

There were a lot of strength and limitation of this project. Talking about the strength of project we were able to work smoothly as a group and understanding as we could discuss about any problems faced with each other. Help cost with different system analysis and interface design field. We also worked on full stack development using MERN Stack which is a much needed plus point for software engineer. We were also able to learn about the database work and its importance on backend and front-end development. The knowledge we got through this project was a very good experience for all of us and will also be helpful all other projects we will be doing in future in our personal life when we will be working for bigger software companies and on bigger projects.

Along with many strength limitations of this project. The main drawback was insufficient project due date. Just a month for full stack development project along with documentation was not enough at all which caused various problems to the development team and other limitation was that not enough information provided by client side during interview session.

Completing project this soon was overall a successful tasks for us. We have completed this project with including all 10 modules and their functionalities. While testing the web application we coded, we found various bugs and errors but eventually we solved all of them and at final revaluation result no errors were found and app was quite good in functionalities. Now what we need to do next is to work more on this type of full stack project and try to provide more functionalities in projects fulfill. All the required needs of client side in future.

# 9. Project Management

## 9.1 Project Gantt Chat/WBDS/Activity List

## 9.2 Project Meeting Minutes

1. **Meeting Minutes**

**Date : 7th May 2022**

**Meeting :** Group Project

**Meeting Duration : 2.**5 hours

**Location :** Virtual via Discord

**Members Present :** Saurab Khadka, Sadikshya Baral, Rohan Niraula, Salon Koirala, Mukesh Singh

**Tutor : Suresh Gautam**

**Project Discussion :** All members discussed about the project and their

  understanding.

**Outcome : Understood all opinions of group member**

1. **Meeting Minutes**

**Date : 8th May 2022**

**Meeting :** Group Project

**Meeting Duration :** 30 minutes

**Location :**  Google Meet

**Members Present :** Saurab Khadka, Sadikshya Baral, Rohan Niraula, Salon Koirala, Mukesh Singh

**Tutor : Suresh Gautam**

**Project Discussion : Interview with the Stake Holders**

**Outcome :** The set of questions were prepared for the Stake Holders

1. **Meeting Minutes**

**Date : 9th May 2022**

**Meeting :** Group Project

**Meeting Duration :** 1 hour

**Location :** Virtual via Discord

**Members Present :** Saurab Khadka, Sadikshya Baral, Rohan Niraula, Salon Koirala, Mukesh Singh

**Tutor : Suresh Gautam**

**Project Discussion :** Divided the work among group member for documentation

**part.**

**Outcome :** divided the work and completion deadline was fixed to

**May 12.**

1. **Meeting Minutes**

**Date : 11th May 2022**

**Meeting :** Group Project

**Meeting Duration :** 2 hours

**Location :** Discord

**Members Present :** Saurab Khadka, Sadikshya Baral, Rohan Niraula, Salon Koirala, Mukesh Singh

**Tutor : Suresh Gautam**

**Project Discussion : Discussed if any member faced any kind of problem while doing documentation of project**

**Outcome :** Showed the ongoing work portion to each other.

1. **Meeting Minutes**

**Date : 13th May 2022**

**Meeting :** Group Project

**Meeting Duration :** 1 hour

**Location :** Discord

**Members Present :** Saurab Khadka, Sadikshya Baral, Rohan Niraula, Salon Koirala, Mukesh Singh

**Tutor : Suresh Gautam**

**Project Discussion : Update till date: Completed the documentation part**

**that is upto system interface design**

**Outcome :** discussed on coding part of project and decided to use

MERN stack for our coding portion.

**6. Meeting Minutes**

**Date : 14th May 2022**

**Meeting :** Group Project

**Meeting Duration :** 2 hours

**Location :** Discord

**Members Present :** Saurab Khadka, Sadikshya Baral, Rohan Niraula, Salon Koirala, Mukesh Singh

**Tutor : Suresh Gautam**

**Project Discussion : Discussed on our learning on MERN Stack**

**Outcome :** Watched various Youtube video and learned basic MERN stack from there.

**7. Meeting Minutes**

**Date : 25th May 2022**

**Meeting :** Group Project

**Meeting Duration :** 2 hours

**Location :** Discord

**Members Present :** Saurab Khadka, Sadikshya Baral, Rohan Niraula,

**Tutor : Suresh Gautam**

**Project Discussion : Discussed about what problems we were facing and how current work is going on**

**Outcome :** Completed coding for 3 modules i.e. student, staff and course

**8.Meeting Minutes**

**Date : 20th May 2022**

**Meeting :** Group Project

**Meeting Duration :** 2 hours

**Location :** Discord

**Members Present :** Saurab Khadka, Sadikshya Baral, Rohan Niraula,

**Tutor : Suresh Gautam**

**Project Discussion :** shared information and techniques about how to code efficiently

**Outcome :** Completed coding for all remaining modules.

1. **Meeting Minutes**

**Date : 9h Apr 2021**

**Meeting :** Group Project

**Meeting Duration :** 2 hours

**Location :** Library

**Members Present :** Saurab Khadka, Sadikshya Baral, Rohan Niraula,

**Tutor : Suresh Gautam**

**Project Discussion : Update till date**

**Completed overall coding part and now working for remaining part of documentation i.e testing, evaluation , conclusion , etc.**

**Outcome :** Discussed on how remaining work is to be done

## 9.3 Project Quality Plan/Strategy

For the project plan and strategy we tried to implement all the functions client wished. We provided all the 10 modules in the course management system.

And incase if there is any missing functionalities in the project, we will consult with the client team and surely will implement those missing functionalities during the maintenance of the system soon.

We have also planned to do regular maintenance of the system so that client or any user won’t face any problem or difficulties while using the system and will monitor the system on regular basis.

With all these work we will try our best to provide high quality system always.

## 9.3 Project Work Log

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tasks | Individual Contributions | | | |  |
| Saurab  Khadka | Sadikshya  Baral | Rohan  Niraula | Mukesh  Singh | Salon Koirala |
| INTRODUCTION | 33.33 | 33.33 | 33.33 | 0 | 0 |
| REQUIREMENT ENGINEERING | | | | |  |
| *Interview Plans and findings* | 30 | 30 | 30 | 5 | 5 |
| *Comparable Software System Review* | 20 | 20 | 20 | 20 | 20 |
| *Development Relevant Legislation* | 20 | 20 | 20 | 20 | 20 |
| *Existing Business Operation* | 20 | 20 | 20 | 20 | 20 |
| *Summary of existing business limitations* | 20 | 20 | 20 | 20 | 20 |
| Functional Requirements | 20 | 20 | 20 | 20 | 20 |
| Performance Requirements | 20 | 20 | 20 | 20 | 20 |
| Design Constraints | 20 | 20 | 20 | 20 | 20 |
| Commercial Constraints | 20 | 20 | 20 | 20 | 20 |
| SYSTEM ANALYSIS & DESIGN | | | | |  |
| Preliminary Design Stages | 35 | 30 | 35 | 0 | 0 00 |
| Detailed Static System Designs | 35 | 30 | 35 | 0 | 0 |
| SYSTEM INTERFACE DESIGNS | | | | |  |
| Draft Interface Designs | 35 | 30 | 35 | 0 | 0 0 |
| DESIGN REVISIONS | 35 | 30 | 35 | 0 | 0 |
| SYSTEM BUILD | | | | |  |
| Coding | 33 | 33 | 33 | 0.5 | 0 .5 |
| Testing | 25 | 25 | 25 | 10 | 5 |
| Project Management | 25 | 25 | 25 | 10 | 5 |
| REPORT CONCLUSION | 25 | 25 | 25 | 10 | 5 |
| Project Documentation and formatting | 25 | 25 | 25 | 10 | 5 |

# References:

* Legislation.gov.uk. 1988. Copyright, Designs and Patents Act 1988. Available at: <https://www.legislation.gov.uk/ukpga/1988/48/contents> [Accessed 11th May 2022].
* Legislation.gov.uk. 1990. Computer Misuse Act 1990. Available at: <https://www.legislation.gov.uk/ukpga/1990/18/contents> [Accessed 11th May 2022].
* Legislation.gov.uk. 2017. Digital Economy Act 2017. Available at: <https://www.legislation.gov.uk/ukpga/2017/30/contents> [Accessed 11th May 2022].
* Legislation.gov.uk. 2018. Data Protection Act 2018. Available at: <http://www.legislation.gov.uk/ukpga/2018/12/part/6/crossheading/jurisdiction-of-courts/enacted> [11th May 2022].
* Npm.2022.react-to-prin *t*. Available at: <https://www.npmjs.com/package/react-to-print> [Accessed 17th May 2022].
* W3schools.com. 2022. *React JSX*. [online] Available at: <https://www.w3schools.com/REACT/react\_jsx.asp> [Accessed 17th May 2022].
* www.javatpoint.com. 2022. *ReactJS Components - javatpoint*. [online] Available at: <https://www.javatpoint.com/react-components> [Accessed 18th May 2022].
* Youtube.com. 2022. [online] Available at: <https://www.youtube.com/watch?v=5Y5QKfxTErU&t=1925s> [Accessed 20 May 2022].
* Youtube.com. 2022. [online] Available at: <https://www.youtube.com/watch?v=ngc9gnGgUdA&t=1739s> [Accessed 21 May 2022].