1. 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.

1. 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)**
    3. **Dr. Raj Singh (Senior lecturer, Module leader, Personal Tutor)**
    4. **Mr. Mark Williams (Existing student)**
    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. |

# 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.

* + - 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.

**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 |

1. 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
     1. 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 |

* + 1. 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  Full Name  Address  Contact  DOB  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  Full Name  Address  Contact  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 |

* + 1. 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**

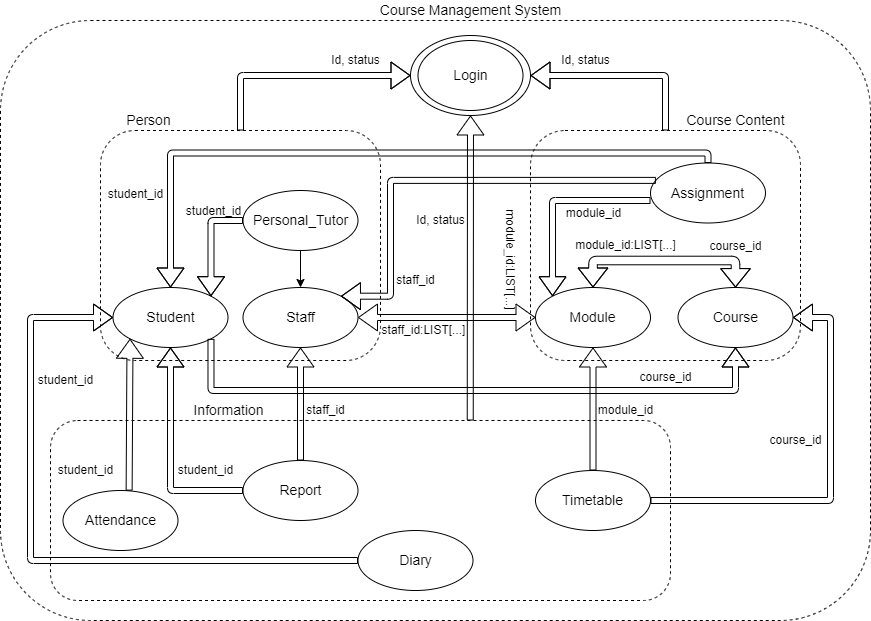
|  |  |  |  |
| --- | --- | --- | --- |
| CLASS | REPORT\_GENERATION | | PART 1/1 |
| TYPE OF OBJECT  A report generation object in the system. | | Author - Saurab Khadka  Created – 9 May, 2022 | |
| Queries | |  | |
| Commands | | Create, Display, Print | |
| Constraints | | * Administrator can access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

* + 1. 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. | |

* + 1. 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. | |

* 1. Detailed Dynamic System Designs
     1. 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

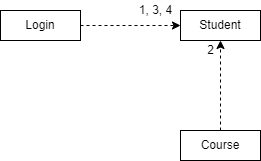
|  |  |  |  |
| --- | --- | --- | --- |
| **EVENTS** | **COURSE\_MANAGEMENT\_SYSTEM** | | **PART 1/1** |
| **COMMENT**  List of classes creating objects in the system. | | Created – 10 May, 2022 | |
| **Class** | | **Creates instances of** | |
| LOGIN | | - | |
| STUDENT | | COURSE | |
| STAFF | | MODULE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

* + 1. Dynamic Diagrams

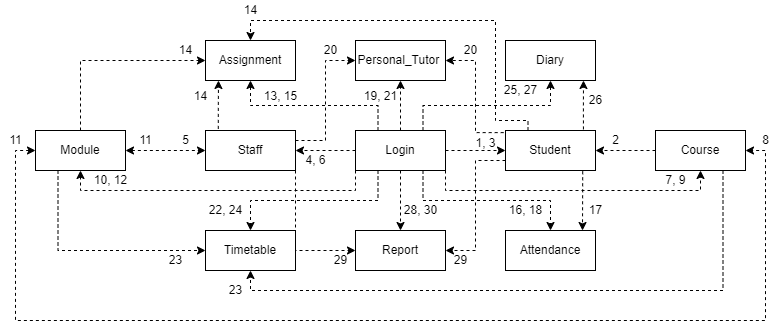
|  |
| --- |
| Scenario:   * + - 1. Open student record       2. Ask course id from course       3. Include course id in student record       4. Display student record |



BON Dynamic Diagram of Student record

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

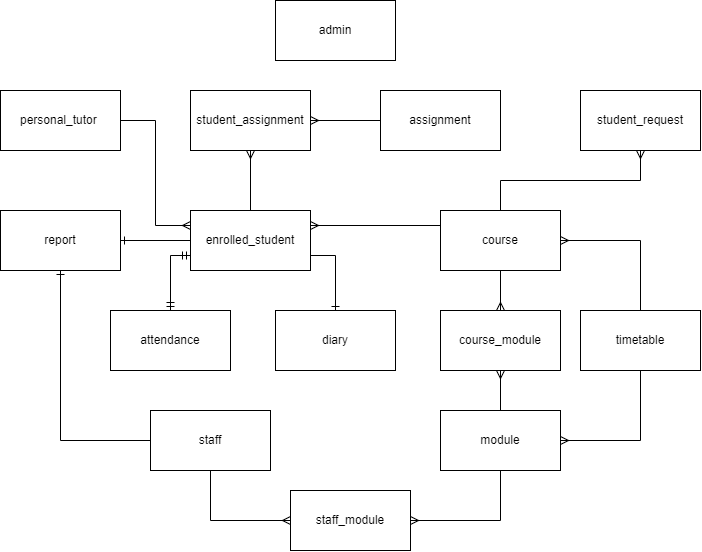
|  |
| --- |
| Scenario:   1. Log into the system and open student. 2. Access course Id from course. 3. Display student. 4. Check the staff. 5. Import module Id from module. 6. Display staff. 7. Open course. 8. Access module list. 9. Display course. 10. Open module. 11. Import staff Id from staff and 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

* 1. 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

* + 1. Attribute Listings

admin

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Datatype | Constraint | Default |
| admin\_id | NUMBER(8) | PK, AUTO INCRE |  |
| password | VARCHAR(20) | NOT NULL |  |

enrolled\_student

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Datatype | Constraint | Default |
| student\_id | NUMBER(8) | PK,AUTO INCRE |  |
| fullname | VARCHAR(50) | NOT NULL |  |
| address | VARCHAR(50) | NOT NULL |  |
| contact | VARCHAR(15) | NOT NULL |  |
| dob | DATE | NOT NULL |  |
| join\_year | VARCHAR(4) | NOT NULL |  |
| section | VARCHAR(5) | NOT NULL |  |
| course\_id\* | NUMBER(8) | FK, NOT NULL |  |
| tutor\_id\* | NUMBER(8) | FK, NOT NULL |  |
| information\_link | VARCHAR(200) | NOT NULL |  |

student\_request

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Datatype | Constraint | Default |
| sn | NUMBER(8) | NOT NULL |  |
| fullname | VARCHAR(50) | NOT NULL |  |
| address | VARCHAR(50) | NOT NULL |  |
| contact | VARCHAR(15) | NOT NULL |  |
| dob | DATE | NOT NULL |  |
| join\_year | VARCHAR(4) | NOT NULL |  |
| course\_id\* | NUMBER(8) | FK, NOT NULL |  |
| information\_link | VARCHAR(200) | NOT NULL |  |

staff

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Datatype | Constraint | Default |
| staff\_id | NUMBER(8) | PK, AUTO INCRE |  |
| fullname | VARCHAR(50) | NOT NULL |  |
| address | VARCHAR(50) | NOT NULL |  |
| contact | VARCHAR(15) | NOT NULL |  |
| module\_id\* | NUMBER(8) | FK, NOT NULL |  |

course

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Datatype | Constraint | Default |
| course\_id | NUMBER(8) | PK, AUTO INCRE |  |
| course\_name | VARCHAR(50) | NOT NULL |  |

module

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Datatype | Constraint | Default |
| module\_id | NUMBER(8) | PK, AUTO INCRE |  |
| module\_name | VARCHAR(50) | NOT NULL |  |

staff\_module

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Datatype | Constraint | Default |
| staff\_id\* | NUMBER(8) | PK, FK, NOT NULL |  |
| module\_id\* | NUMBER(8) | PK, FK, NOT NULL |  |

course\_module

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Datatype | Constraint | Default |
| course\_id\* | NUMBER(8) | PK, FK, NOT NULL |  |
| module\_id\* | NUMBER(8) | PK, FK, NOT NULL |  |

assignment

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Datatype | Constraint | Default |
| assignment\_id | NUMBER(8) | PK,AUTO INCRE |  |
| assignment\_name | VARCHAR(50) | NOT NULL |  |
| module\_id\* | NUMBER(8) | FK, NOT NULL |  |
| initialized\_date | DATE | NOT NULL |  |
| deadline | DATE | NOT NULL |  |
| information | VARCHAR(255) |  |  |

student\_assignment

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Datatype | Constraint | Default |
| student\_id\* | NUMBER(8) | PK, FK |  |
| assignment\_id\* | NUMBER(8) | PK, FK |  |
| submission\_link | VARCHAR(200) |  |  |

attendance

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Datatype | Constraint | Default |
| student\_id\* | NUMBER(8) | FK |  |
| student\_name | VARCHAR(50) | NOT NULL |  |
| year | VARCHAR(4) | NOT NULL |  |
| total\_present\_days | NUMBER(3) | NOT NULL |  |
| possible\_present\_days | NUMBER(3) | NOT NULL |  |
| attendance\_percent | NUMBER(3) | NOT NULL |  |

personal\_tutor

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Datatype | Constraint | Default |
| staff\_id\* | NUMBER(8) | FK, NOT NULL |  |
| student\_id\* | NUMBER(8) | FK, NOT NULL |  |

timetable

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Datatype | Constraint | Default |
| course\_id\* | NUMBER(8) | FK, NOT NULL |  |
| day | VARCHAR(10) | NOT NULL |  |
| time | TIME | NOT NULL |  |
| module\_id\* | NUMBER(8) | FK |  |

report

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Datatype | Constraint | Default |
| student\_id\* | NUMBER(8) | FK |  |
| staff\_id\* | NUMBER(8) | FK |  |
| report | BINARY(8000) |  |  |

diary

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Datatype | Constraint | Default |
| id | NUMBER(8) | PK |  |
| message | VARCHAR(255) |  |  |
| schedule | VARCHAR(200) |  |  |
| student\_id\* | NUMBER(8) | FK |  |

* 1. Detailed Dynamic System Designs

1. System Interface Design

4.1 Draft Interface Design (Record Management)

4.1.1 Wireframes

Below are the wireframes of the Record Management.

4.1.2 System Navigation Diagram (Record Management)

4.1.3 System Screen Mock-ups (Record Management)

|  |
| --- |
| Graphical user interface  Description automatically generated with medium confidence  Figure 2 Login Page of Record Management System |

|  |
| --- |
| Graphical user interface, application  Description automatically generated  Figure 3 Non enrolled Section of Record Management System |

|  |
| --- |
| Graphical user interface, application  Description automatically generated  Figure 4 Enrolled Section of Student in Record Management System |

|  |
| --- |
| A picture containing graphical user interface  Description automatically generated  Figure 5 Archived Student Date in Record Management System |

|  |
| --- |
| Graphical user interface, application, Teams  Description automatically generated  Figure 6 Create Student data in Record Management System |

|  |
| --- |
| Graphical user interface, application, Teams  Description automatically generated  Figure 7 Update Student Data in Record Management System |

|  |
| --- |
| A screenshot of a computer  Description automatically generated with medium confidence  Figure 8 Staff Data in Record Management System |

|  |
| --- |
| A picture containing graphical user interface  Description automatically generated  Figure 9 Archived Staff Date in Record Management System |

|  |
| --- |
| Graphical user interface, application, Teams  Description automatically generated  Figure 10 Update Staff Record in Record Management System |

|  |
| --- |
| A screenshot of a computer  Description automatically generated with medium confidence  Figure 11 Course Section in Record Management System |

|  |
| --- |
| A picture containing graphical user interface  Description automatically generated  Figure 12Archived Course in Record Management System |

|  |
| --- |
| Graphical user interface, application, Teams  Description automatically generated  Figure 13 Create Course Sturcture in Record Management System |

|  |
| --- |
| Graphical user interface, application, Teams  Description automatically generated  Figure 14 Update Course Structure in Record Management System |

|  |
| --- |
| A screenshot of a computer  Description automatically generated with medium confidence  Figure 15 Module Section in Record Management System |

|  |
| --- |
| Graphical user interface  Description automatically generated with low confidence  Figure 16 Archived Module data in Record Management System |

|  |
| --- |
| A picture containing shape  Description automatically generated  Figure 17 Create Module record in Record Management System |

|  |
| --- |
| Shape  Description automatically generated with low confidence  Figure 18 Update Module Record in Record Management System |

|  |
| --- |
| A screenshot of a computer  Description automatically generated with medium confidence  Figure 19 Assignment Section in Record Management System |

|  |
| --- |
| A screenshot of a computer  Description automatically generated with medium confidence  Figure 20 Assignment Section when clicked in particular Assignment |

|  |
| --- |
| Graphical user interface, application, Teams  Description automatically generated  Figure 21 Assign Assignment in Record Management System |

|  |
| --- |
| Graphical user interface, application  Description automatically generated  Figure 22 Mark Student Submitted Assignment in Record Management System |

|  |
| --- |
| A picture containing graphical user interface  Description automatically generated  Figure 23 Archived Assignment in Record Management System |

|  |
| --- |
| Graphical user interface, text, application, chat or text message  Description automatically generated  Figure 24 Attendance Section in Record Management System |

|  |
| --- |
| A picture containing graphical user interface  Description automatically generated  Figure 25 Archived Attendance in Record Management System |

|  |
| --- |
| Graphical user interface, application, Teams  Description automatically generated  Figure 26 Create Attendance Sheet section in Record Management System |

|  |
| --- |
| Graphical user interface, application, Teams  Description automatically generated  Figure 27 Update Attendance Sheet in Record Management System |

|  |
| --- |
| A picture containing table  Description automatically generated  Figure 28 Take Attendance Sheet in Record Management System |

|  |
| --- |
| A screenshot of a computer  Description automatically generated with medium confidence  Figure 29 Personal Tutor Section in Record Management System |

|  |
| --- |
| A picture containing graphical user interface  Description automatically generated  Figure 30 Archived Personal Tutor in Record Management System |

|  |
| --- |
| Graphical user interface, application, Teams  Description automatically generated  Figure 31 Edit Personal Tutor Section in Record Management System |

|  |
| --- |
| Table  Description automatically generated  Figure 32 Assign Personal Tutor to Student Section in Record Management System |

|  |
| --- |
| A screenshot of a computer  Description automatically generated with medium confidence  Figure 33 Timetable Section in Record Management System |

|  |
| --- |
| A picture containing graphical user interface  Description automatically generated  Figure 34 Archived Timetable in Record Management System |

|  |
| --- |
| Table  Description automatically generated  Figure 35 Create Timetable Section in Record Management System |

|  |
| --- |
| Table  Description automatically generated  Figure 36 Update Timetable Section in Record Management System |

|  |
| --- |
| Graphical user interface, application  Description automatically generated  Figure 37 Diary Section in Record Management System |

|  |
| --- |
| A picture containing graphical user interface  Description automatically generated  Figure 38 Archived Diaries in Record Management System |

|  |
| --- |
| Graphical user interface, application, Teams  Description automatically generated  Figure 39 Create Diary Section in Record Management System |

|  |
| --- |
| Graphical user interface, application, Teams  Description automatically generated  Figure 40 Update Diary in Record Management System |

|  |
| --- |
| Diagram  Description automatically generated  Figure 41 Generate Report Section in Record Management System |

|  |
| --- |
| Shape, rectangle  Description automatically generated  Figure 42 View Report Section in Record Management System |

|  |
| --- |
|  |

|  |
| --- |
|  |

4.1.4 System Activity Event Diagram (Record Management)

4.2 Design Revisions (Record Management)

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

4.3 Heuristic Evaluation (Records Management)

4.4 Draft Interface Design (Student Record/Information Portal)

4.4.1 Wireframes

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

4.4.2 System Navigation Diagram (Student Record/Information Portal)

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

|  |
| --- |
| Graphical user interface  Description automatically generated  Figure 43 Courses section in Student Portal |

|  |
| --- |
| Graphical user interface  Description automatically generated with medium confidence  Figure 44 Module Section in Student Portal |

|  |
| --- |
| Graphical user interface, application  Description automatically generated  Figure 45 Module Material Section in Student Portal |

|  |
| --- |
| Graphical user interface  Description automatically generated  Figure 46 Assignment Section in Student Portal |

|  |
| --- |
| Graphical user interface, text, application  Description automatically generated  Figure 47 Submit Assignment Section in Student Portal |

|  |
| --- |
| Graphical user interface, application  Description automatically generated  Figure 48 Personal Tutor Section in Student Portal |

|  |
| --- |
| Text  Description automatically generated  Figure 49 TimeTable Section in Student Portal |

|  |
| --- |
| Graphical user interface, text  Description automatically generated with medium confidence  Figure 50 Diary Section in Student Portal |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

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

4.5 Design Revisions (Student Record/Information Portal)

The design above presented are the belonging of the Record Management

4.6 Heuristic Evaluation (Student Record/Information Portal)