* 1. **Project Development Methodology**

The project will be developed using an agile software development methodology, which is well-suited to mobile app development projects. The agile methodology is characterized by a flexible and iterative approach to software development, which allows for changes to be made to the project requirements and design as needed.

The development process will be divided into a series of sprints, each lasting approximately two weeks. At the start of each sprint, the development team will meet to review the project requirements and design, and to plan the work to be carried out during the sprint. The team will work closely together during the sprint to develop and test the app features, and to resolve any issues that arise.

Regular meetings will be held between the development team and the project stakeholders to review progress, discuss any issues, and make decisions about the project direction. The project will be managed using project management software, which will enable the project team to track progress, manage tasks, and communicate effectively throughout the development process.

1. **Requirement Engineering**

**2.1 Elicitation Activities**

The purpose of this report is to outline the interview plans for the requirements engineering process. The requirements engineering process is an important phase of software development where the needs and requirements of stakeholders are identified and analyzed. Interviews are a primary method of eliciting requirements from stakeholders, and effective interview planning is necessary for a successful requirements engineering process. This report will provide an overview of the interview plans for the requirements engineering process.

**2.1.1 Interview Plan:**

The interview plan is an important aspect of the requirements engineering process. It outlines the approach that will be taken during the interview process and provides a framework for the interviewer to follow. The following are the key components of the interview plan:

1. Identify stakeholders: The first step in the interview planning process is to identify the stakeholders who will be interviewed. This includes individuals who have a vested interest in the software system, such as end-users, developers, project managers, and business owners.
2. Define the objectives: The objectives of the interview should be clearly defined before the interview process begins. This includes understanding the needs, goals, and requirements of the stakeholders and how they relate to the software system.
3. Develop the interview questions: The interview questions should be carefully developed to ensure that they are relevant to the stakeholder's objectives. This includes open-ended questions that allow stakeholders to provide detailed responses and closed-ended questions that provide specific information.
4. Schedule the interview: The interview schedule should be carefully planned to ensure that stakeholders are available and that the interview does not interfere with their work.
5. Conduct the interview: The interviewer should follow the interview plan and ask the questions that have been developed. The interviewer should also be prepared to ask follow-up questions to clarify the stakeholder's responses.
6. Record the interview: The interview should be recorded to ensure that all the information is captured accurately. The interviewer should also take notes during the interview to capture any relevant information that is not captured in the recording.

Conclusion: Effective interview planning is essential for the success of the requirements engineering process. The interview plan should be carefully developed to ensure that all the key stakeholders are identified, the objectives of the interview are clearly defined, and the interview questions are relevant to the stakeholder's objectives. The interview plan should be followed carefully to ensure that all relevant information is captured accurately, and the interviewer should be prepared to ask follow-up questions to clarify the stakeholder's responses.

**2.2 Requirements Specification**

2.2.1 Problem Domain Description: In this scenario, the problem domain is the education sector, specifically focusing on the student lifecycle and personal tutorial lifecycle within an existing business operation. The objective is to optimize these processes and enhance the overall experience of students.

**2.2.1.1 Existing Business Operation:**

The existing business operation is an educational institution that offers courses and programs to students. It provides a range of services to support students throughout their academic journey. These services include admissions, registration, scheduling, grading, and graduation.

**2.2.1.1.1 Student Lifecycle:**

The student lifecycle refers to the entire process that a student goes through, from initial inquiry to graduation. It involves several stages such as admissions, enrollment, course selection, attendance tracking, grading, and graduation. The institution needs to manage these stages efficiently and effectively to ensure that students receive a high-quality education and have a positive experience.

**2.2.1.1.2 Personal Tutorial Lifecycle:**

The personal tutorial lifecycle is a subset of the student lifecycle that focuses on providing one-on-one support to students. This support can be in the form of academic tutoring, career counseling, or personal development. The institution needs to ensure that students have access to personalized support throughout their academic journey to help them achieve their goals.

**2.2.1.1.3 Optional Process 1:**

An optional process that the institution may offer is an internship or co-op program. This program provides students with the opportunity to gain practical work experience while still studying. The institution needs to ensure that students are well-prepared for these programs, that they are matched with appropriate employers, and that they receive support during their internship or co-op.

**2.2.1.1.4 Optional Process 2:**

Another optional process that the institution may offer is a study abroad program. This program provides students with the opportunity to study in a foreign country, learn about different cultures, and gain new perspectives. The institution needs to ensure that students are well-prepared for this experience, that they are matched with appropriate universities, and that they receive support during their study abroad program

**2.2.2 Functional Requirements**

There are three main functional requirements for the Woodlands University College project:

1. Records Management Systems: The records management system should be able to manage and store student records, personal tutor records, and work experience placement records. It should also be able to generate reports on the data stored.
2. Student Records/Information Portal: The student information portal should allow students to access their academic records and personal information, view and manage their work experience placement information, and communicate with their personal tutors.
3. Woodlands University College Corporate Website: The corporate website should provide information about the university, its courses, staff, and other important information, including news and events.

**2.2.3 Performance Requirements**

To meet the performance requirements of the Woodlands University College project, the following criteria should be met:

**2.2.3.1 Records Management:**

**2.2.3.1.1 Speed**

The records management system must be able to quickly and efficiently retrieve and store student records. All search functions must provide results within 2 seconds. The system must be able to handle a large number of concurrent user requests without slowing down the performance. All data input and output operations must be completed within 3 seconds.

**2.2.3.1.2 Capacity**

The system must be able to handle large volumes of data storage and retrieval, accommodating for future growth. The system must be designed to handle a large number of concurrent user requests. The system must have enough storage capacity to handle all required student records.

**2.2.3.1.3 Reliability**

The system must be rigorously tested to ensure that there are no errors or bugs. The system must be designed to handle errors and prevent data loss. The system must have a backup and recovery plan in case of data loss or system failure. The system must have a secure login system to prevent unauthorized access.

**2.2.3.1.4 Usability**

The system must be easy to navigate and user-friendly. The system must have clear and concise instructions for all functions and processes. The system must be designed to minimize the risk of user errors. The system must have a user-friendly interface for inputting and retrieving student records.

**2.2.3.1.5 Accessibility**

The system must be accessible to authorized users from any location with an internet connection. The system must have accessibility features for users with disabilities, such as screen readers and keyboard shortcuts. The system must comply with accessibility standards, such as Section 508 of the Rehabilitation Act and the Web Content Accessibility Guidelines (WCAG).

**2.2.3.2 Student Records/Information Portal**

**2.2.3.2.1 Speed**

The portal must load all pages within 3 seconds. All data retrieval operations on the portal must be completed within 5 seconds. All forms submissions must be completed within 5 seconds. The portal must provide fast and efficient search functionality to quickly retrieve records.

**2.2.3.2.2 Capacity**

The portal must be designed to handle a large number of concurrent users without any loss of performance. The database must be optimized to handle a large volume of records and should be able to handle 10,000 users simultaneously.

**2.2.3.2.3 Reliability**

The portal must be tested thoroughly to remove any errors and bugs. The data must be backed up regularly to avoid any data loss in the event of a system failure or disaster. The portal must have a failsafe mechanism that can handle unexpected errors and prevent any loss of data.

**2.2.3.2.4 Usability**

The portal must be user-friendly and easy to navigate.The portal should have an intuitive user interface that is easy to use for all types of users. The portal should provide clear instructions and guidance on how to use its features. The portal should provide users with the ability to view and edit their personal information. The portal should provide users with easy access to all of their academic records, including grades, attendance, and course schedules.

**2.2.3.2.5 Accessibility**

The portal must be accessible to all users, including those with disabilities. The portal should comply with web accessibility standards, such as WCAG 2.0, to ensure that users with disabilities can access all of its features. The portal should provide alternative text descriptions for all images and multimedia content. The portal should be designed to work with assistive technologies, such as screen readers and voice recognition software.

**2.2.3.3 Woodlands University College Corporate Website**

**2.2.3.3.1 Speed**

All pages in the Woodlands University College corporate website should load in less than 3 seconds to provide a good user experience. All web page postbacks should be completed in less than 3 seconds to avoid user frustration.

**2.2.3.3.2 Capacity**

The website should be able to handle at least 5000 concurrent users without significantly impacting the system performance.

**2.2.3.3.3 Reliability**

The website should be available 24/7 with a maximum of 1 hour of downtime per week for maintenance. All data stored in the website should be backed up regularly to ensure data is not lost in case of system failure. In case of system failure, the system should be able to recover data up to the point of failure without data loss.

**2.2.3.3.4 Usability**

The Woodlands University College corporate website should have an intuitive and user-friendly interface that is easy to navigate. The interface should have clear and concise instructions for each function. The interface should provide adequate feedback to the user after each operation, including error messages if necessary. The interface should be accessible to users with disabilities, including those who use assistive technologies such as screen readers.

**2.2.3.3.5 Accessibility**

The Woodlands University College corporate website should be accessible from a variety of devices and platforms, including desktop and mobile devices. The website should be compatible with the latest versions of major web browsers, including Chrome, Firefox, Safari, and Edge. The website should be designed with web accessibility guidelines in mind, including WCAG 2.1, to ensure it is accessible to users with disabilities.

**2.2.3.4 Woodlands University College Mobile Application – Student Information**

**2.2.3.4.1 Speed**

The mobile application must load within 3 seconds. The data retrieval and processing time must be less than 2 seconds. The time taken to switch between screens or modules of the application must be less than 1 second. The mobile application should have efficient data synchronization with the backend server to minimize data transfer time.

**2.2.3.4.2 Capacity**

The mobile application must support at least 10,000 concurrent users. The application must handle a large amount of data including class schedules, grades, and attendance records.

**2.2.3.4.3 Reliability**

The application must be rigorously tested to remove any errors. The application must work seamlessly in different network conditions (3G, 4G, Wi-Fi, etc.). The application should have proper error handling mechanisms and display informative messages in case of any error. The application should have a mechanism to save user progress, in case of any unexpected shutdown or interruption.

**2.2.3.4.4 Usability**

The application must be easy and intuitive to navigate. The application should be designed with a user-friendly interface to enable easy access to student information. All functional features of the system that update or delete student data must give the user a chance to confirm or cancel the action. Following all actions that create, update, or delete student data, a message must be displayed informing the user of the action's success. The mobile application must have a feature to provide push notifications for important events such as class cancellations or changes in schedule.

**2.2.3.4.5 Accessibility**

The application must be accessible to users with disabilities. The application must support different font sizes and screen contrasts. The mobile application should support multiple languages for international students.

**2.2.4 Design Constraints**

To ensure the Woodlands University College project meets design constraints, the following requirements should be met:

**2.2.4.1 Records Management Systems:**

* Must comply with relevant data protection and privacy regulations
* Must have secure login and authentication system for authorized access
* Must have backup and disaster recovery procedures in place
* Must be able to handle a large volume of records efficiently
* Must have search and filtering capabilities for quick access to specific records
* Must be able to generate various reports and statistical data on request
* Must have access control and permission settings for different user roles
* Must be able to integrate with other systems used by the university

**2.2.4.2 Student Records/Information Portal:**

* Must have a user-friendly interface for easy navigation and accessibility
* Must have secure login and authentication system for authorized access
* Must display accurate and up-to-date student information and records
* Must provide options for students to update their personal information
* Must have search and filtering capabilities for quick access to specific information
* Must provide access to course information, schedules, grades, and financial information
* Must provide notifications for important events and deadlines
* Must be mobile-responsive for access on different devices
* Must integrate with other university systems such as the Records Management System

**2.2.4.3 Woodlands University College Corporate Website:**

* Must have a professional and modern design that reflects the university's brand
* Must have clear and concise information about the university's mission, programs, and services
* Must be easy to navigate and find information
* Must be accessible to people with disabilities
* Must have a responsive design for access on different devices
* Must comply with web standards and regulations
* Must have search engine optimization (SEO) to improve visibility and ranking on search engines
* Must be regularly updated with fresh content and news
* Must provide options for feedback and inquiries

**2.2.4.4 Woodlands University College Mobile Application – Student Information:**

* Must have a user-friendly interface for easy navigation and accessibility
* Must have secure login and authentication system for authorized access
* Must provide accurate and up-to-date student information and records
* Must provide options for students to update their personal information
* Must have search and filtering capabilities for quick access to specific information
* Must provide access to course information, schedules, grades, and financial information
* Must provide notifications for important events and deadlines
* Must be optimized for mobile devices and have a responsive design
* Must be able to work offline and synchronize data when connected
* Must integrate with other university systems such as the Student Records/Information Portal

**2.2.5 Commercial Constraints (Total Project)**

The total project should be completed within a budget of £250,000, including the costs of software development, hardware, and personnel costs.

**2.2.6 Acceptance Tests (Total Project)**

Acceptance tests must be performed to ensure that the system meets the functional and performance requirements specified in this document. The acceptance tests must be designed to test the system in various scenarios, including high traffic periods, system failures, and security breaches. The acceptance tests must be documented, and the development team must ensure that all test cases are passed before the system is released to production.

Conclusion

This document has outlined the requirements for the development of a new student records management system for Woodlands University College. The system must meet certain functional and performance requirements and adhere to specific design and commercial constraints. Additionally, the system must pass acceptance tests to ensure that it meets the needs of the college and its students

**3.1 Preliminary Design Stages:**

**3.1.1 Textual Analysis**

In this stage, the project team analyzes the requirements documentation and other relevant information to identify the needs and goals of the system. This involves reviewing and clarifying the requirements, identifying any missing or ambiguous information, and generating a list of system features and functions.

|  |  |
| --- | --- |
| **Candidate class** | Candidate Routines |
| Student  Course  Instructor  Assignment  Exam  Grade  Department  Academic Record  Schedule  Classroom  Resource (e.g. textbooks, equipment)  Notification | Amend  Archive  Assign  Create  Delete  Display  Enroll  Grade  Notify  Print  Read  Search  Sign-in  Add |

**3.1.2 Significant Event Analysis**

This stage involves identifying the key events and interactions that occur within the system, as well as the actors involved in those events. This helps to define the scope of the system and clarify the major components and functions.

|  |  |  |
| --- | --- | --- |
| **Event** | **Performers** | **Candidate Attributes** |
| Enroll Student | Student  Administration Staff Member | * Student ID number * Student First Name * Student Surname * Student E-mail * Student Telephone No * Student Address * Student Level of Study |
| Drop Student | Student  Administration Staff Member | Student ID number |
| Add Course | Administration Staff Member | * Course ID * Course Name |
| Edit Course | Administration Staff Member | * Course ID * Course Name |
| Remove Course | Administration Staff Member | Course ID |
| Add Instructor | Administration Staff Member | * Staff Name * Staff ID |
| Edit Instructor | Administration Staff Member | * Staff Name * Staff ID |
| Remove Instructor | Administration Staff Member | Staff ID |
| Add Assignment | Instructor | * Assignment ID * Assignment Name * Assignment Due Date |
| Edit Assignment | Instructor | * Assignment ID * Assignment Name * Assignment Due Date |
| Remove Assignment | Instructor | Assignment ID |
| Enter Grade | Instructor | * Student ID number * Assignment ID * Exam ID * Grade |
| Add Department | Administration Staff Member | * Department ID * Department Name |
| Edit Department | Administration Staff Member | * Department ID * Department Name |
| Remove Department | Administration Staff Member | Department ID |
| View Academic Record | Student | Student ID number |
| Add Schedule | Administration Staff Member | * Schedule ID * Schedule Date * Schedule Time |
| Edit Schedule | Administration Staff Member | * Schedule ID * Schedule Date * Schedule Time |
| Remove Schedule | Administration Staff Member | Schedule ID |
| Add Classroom | Administration Staff Member | * Classroom ID * Classroom Location |
| Edit Classroom | Administration Staff Member | * Classroom ID * Classroom Location |
| Remove Classroom | Administration Staff Member | Classroom ID |
| Add Resource | Administration Staff Member | * Resource ID * Resource Name * Resource Type |
| Edit Resource | Administration Staff Member | * Resource ID * Resource Name * Resource Type |
| Remove Resource | Administration Staff Member | Resource ID |
| Send Notification | Administration Staff Member | * Notification ID * Notification Recipients * Notification |

**3.1.3 Commands Queries and Constraints**

In this stage, the project team identifies the commands (actions), queries (requests for information), and constraints (rules and limitations) that will be required by the system.

**1.**

|  |  |  |  |
| --- | --- | --- | --- |
| **CLASS** | **Student** | | Part: 1/16 |
| TYPE OF OBJECT  **Student enroll on the course** | | **INDEXING**  Cluster:  **Desktop Application**  Created:  04-03-2023 | |
| Queries | * Student ID number, Student First Name, Student Surname, Student E-mail, Student Telephone No, Student Address, Student Level of Study | | |
| Commands | Create, Amend, Archive, Display, Assign, Print, Search | | |
| Constraints | * • A student cannot enroll in a course that has already started or is full * • A student cannot withdraw from a course after the withdrawal deadline * • An instructor cannot be assigned to a course that conflicts with their schedule * • An assignment's due date cannot be before the date it was assigned * • An exam's date cannot conflict with another exam or course schedule * • A student cannot graduate unless they have completed all required courses and have a minimum GPA * • A student cannot register for more than the maximum number of credits allowed per semester * • A student cannot pay less than the amount due on their account * • A classroom cannot be double-booked * • A resource cannot be reserved for a time period when it is already in use. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **CLASS** | **Course** | | Part: 2/16 |
| TYPE OF OBJECT  **Course information management** | | **INDEXING**  Cluster:  **Desktop Application**  Created:  04-03-2023 | |
| Queries | Course ID, Course Name, Course Description, Course Instructor, Course Schedule, Course Location, Course Credit Hours, Course Prerequisites | | |
| Commands | Create, Amend, Archive, Display, Assign, Print, Search | | |
| Constraints | • A course cannot be created with the same course ID as an existing course  • A course cannot be assigned an instructor who is not qualified to teach that course  • A course cannot be offered if it conflicts with another course in the same department  • A course cannot be offered if it requires a resource that is not available  • A course cannot be assigned a classroom that is too small for the number of students enrolled in the course  • A course cannot be assigned a schedule that conflicts with the instructor's schedule or the schedules of enrolled students • A course cannot be removed from the system if it has active enrollments  • A course cannot be amended after it has started | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **CLASS** | **Instructor** | | Part: 3/16 |
| TYPE OF OBJECT  **Instructor assign to course** | | **INDEXING**  Cluster:  **Desktop Application**  Created:  04-03-2023 | |
| Queries | Staff Name, Staff ID, Staff Email, Staff Telephone No, Staff Department | | |
| Commands | Create, Amend, Archive, Display, Assign, Print, Search | | |
| Constraints | • An instructor cannot be assigned to a course that conflicts with their schedule  • An instructor cannot be assigned to a course for which they are not qualified  • An instructor cannot be assigned to a course that has already started or ended  • An instructor cannot be assigned to a course that has more teaching hours than their contract allows. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **CLASS** | **Assignment** | | Part: 4/16 |
| TYPE OF OBJECT  **Assignment submission** | | **INDEXING**  Cluster:  **Desktop Application**  Created:  04-03-2023 | |
| Queries | Assignment ID, Assignment Name, Assignment Description, Assignment Due Date, Course ID, Instructor Name, Submission Date, Submission Status, Submission ID, Student ID, Student Name, Submission File, Submission Grade | | |
| Commands | Create, Amend, Archive, Display, Assign, Print, Search | | |
| Constraints | • An assignment's due date cannot be before the date it was assigned  • An assignment cannot be submitted after the submission deadline  • An assignment cannot be graded until it has been submitted  • An assignment cannot be graded by the student who submitted it.  • A submission cannot be graded more than once  • A submission's file must meet the file format requirements  • A submission's grade must be within the range of 0 to 100  • An instructor cannot grade an assignment that conflicts with their schedule | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **CLASS** | **Grade** | | Part: 5/15 |
| TYPE OF OBJECT  **Student grade assignment** | | **INDEXING**  Cluster:  **Desktop Application**  Created:  04-03-2023 | |
| Queries | Grade ID number, Assignment ID number, Student ID number, Instructor ID number, Grade Value, Grade Date | | |
| Commands | Create, Amend, Archive, Display, Assign, Print, Search | | |
| Constraints | • A grade cannot be assigned before the due date of the corresponding assignment  • A grade cannot be assigned by an instructor who is not assigned to the corresponding course  • A grade cannot be assigned for an assignment that has been withdrawn or cancelled  • A grade cannot be assigned for a student who has withdrawn from the corresponding course. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **CLASS** | **Department** | | Part: 6/15 |
| TYPE OF OBJECT  **Department Management** | | **INDEXING**  Cluster:  **Desktop Application**  Created:  04-03-2023 | |
| Queries | Department ID, Department Name, Department Location, Department Contact Person, Department Telephone No | | |
| Commands | Create, Amend, Archive, Display, Assign, Print, Search | | |
| Constraints | • A department cannot be deleted if it has active courses or students assigned to it  • Only authorized staff members can create, amend or archive departments  • A department's contact person must be a staff member  • A department's telephone number must be a valid phone number  • A department cannot be assigned to a course that is not offered by the department  • A department cannot be assigned to a course that is already assigned to another department. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **CLASS** | **Academic Record** | | Part: 9/16 |
| TYPE OF OBJECT  **Student academic record** | | **INDEXING**  Cluster:  **Desktop Application**  Created:  04-03-2023 | |
| Queries | Student ID number, Student First Name, Student Surname, Student E-mail, Student Telephone No, Student Address, Student Level of Study, Academic Standing, Completed Courses and Grades, GPA | | |
| Commands | Create, Amend, Archive, Display, Assign, Print, Search | | |
| Constraints | • A student's academic record cannot be altered once they have graduated  • A student's academic record can only be amended by authorized staff members  • A student's academic standing is updated automatically based on their grades and credits completed  • A student's GPA is calculated automatically based on their grades and credits completed | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **CLASS** | **Schedule** | | Part: 13/16 |
| TYPE OF OBJECT  **Scheduling of Courses, Assignments, and Exams** | | **INDEXING**  Cluster:  **Desktop Application**  Created:  04-03-2023 | |
| Queries | Course id, course name, instructor id, instructor name, classroom id, exam id, assignment id | | |
| Commands | Create, Amend, Archive, Display, Print, Search | | |
| Constraints | • A course cannot be scheduled in a time slot that conflicts with another course  • An instructor cannot be assigned to teach two courses at the same time  • An assignment or exam cannot be scheduled in a time slot that conflicts with another assignment, exam, or course  • All courses must be scheduled with a classroom that can accommodate the expected number of students  • All assignments and exams must be scheduled with a time and date that allows students enough time to complete them before the due date  • Any changes made to the schedule must be communicated to all affected parties in a timely manner. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **CLASS** | **Classroom** | | Part: 14/16 |
| TYPE OF OBJECT  **Classroom booking** | | **INDEXING**  Cluster:  **Desktop Application**  Created:  04-03-2023 | |
| Queries | Classroom number, Classroom location, Classroom capacity, Classroom features and amenities, Classroom availability, Classroom schedule | | |
| Commands | Create, Amend, Archive, Display, Assign, Print, Search | | |
| Constraints | • A classroom cannot be double-booked  • A classroom cannot be assigned to a course that conflicts with another course scheduled in the same classroom  • A classroom cannot be assigned to a course that exceeds its maximum capacity  • A classroom must have all the required features and amenities for a course assigned to it  • A classroom's availability must be updated in real-time to prevent conflicts in scheduling | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **CLASS** | **Resource** | | Part: 15/16 |
| TYPE OF OBJECT  **Physical Object** | | **INDEXING**  Cluster:  **Desktop Application**  Created:  04-03-2023 | |
| Queries | * Resource ID number, Resource Name, Resource Description, Resource Type, Resource Availability | | |
| Commands | Create, Update, delete, assign, reserve | | |
| Constraints | * • A resource cannot be deleted if it is currently assigned or reserved * • A resource cannot be assigned or reserved if it is not available * • A resource can only be assigned or reserved for a specific time period * • A resource can only be assigned to one user at a time * • A resource cannot be assigned or reserved for a time period that conflicts with another reservation or assignment. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **CLASS** | **Notification** | | Part: 16/16 |
| TYPE OF OBJECT  **Send notifications to students, instructors, and staff members** | | **INDEXING**  Cluster:  **Desktop Application**  Created:  04-03-2023 | |
| Queries | Notification ID, Message content, Recipient, Sender, Notification type, Notification time, Notification status | | |
| Commands | Send, View, Archive, Delete | | |
| Constraints | • A notification cannot be sent to a non-existent recipient  • A notification cannot be sent if the sender does not have the appropriate permissions  • A notification cannot be sent if it contains inappropriate content  • A notification cannot be deleted once it has been read by the recipient  • A notification cannot be sent for a course that has already ended | | |

**3.2 Detailed Static System Designs:**

**3.2.1 First Draft BON System Architecture Diagram**

A BON (Business Object Notation) system architecture diagram is a high-level view of the system that shows the major components and their relationships. This diagram helps to define the overall structure of the system.

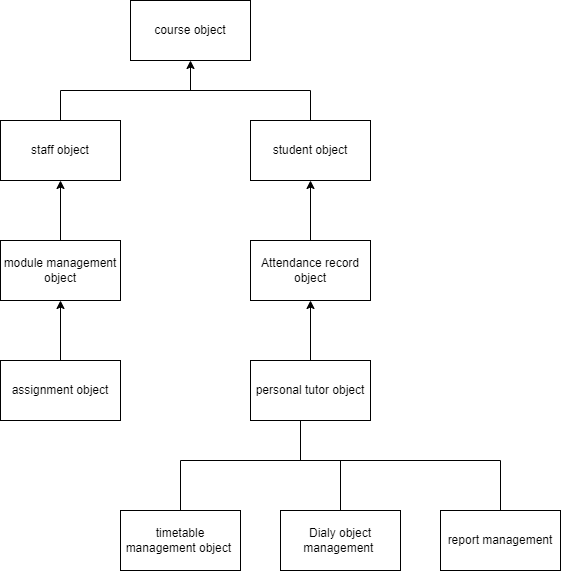


Fig: Bon System Architecture for course management system

The diagram displayed depicts the primary entities and their connections within the course management system. This architectural diagram of the system provides an overview of its structure, allowing stakeholders to comprehend how the various components interrelate.

**3.2.2 BON System Chart**

This chart shows the key objects in the system and their relationships. It helps to define the roles and responsibilities of the different objects.

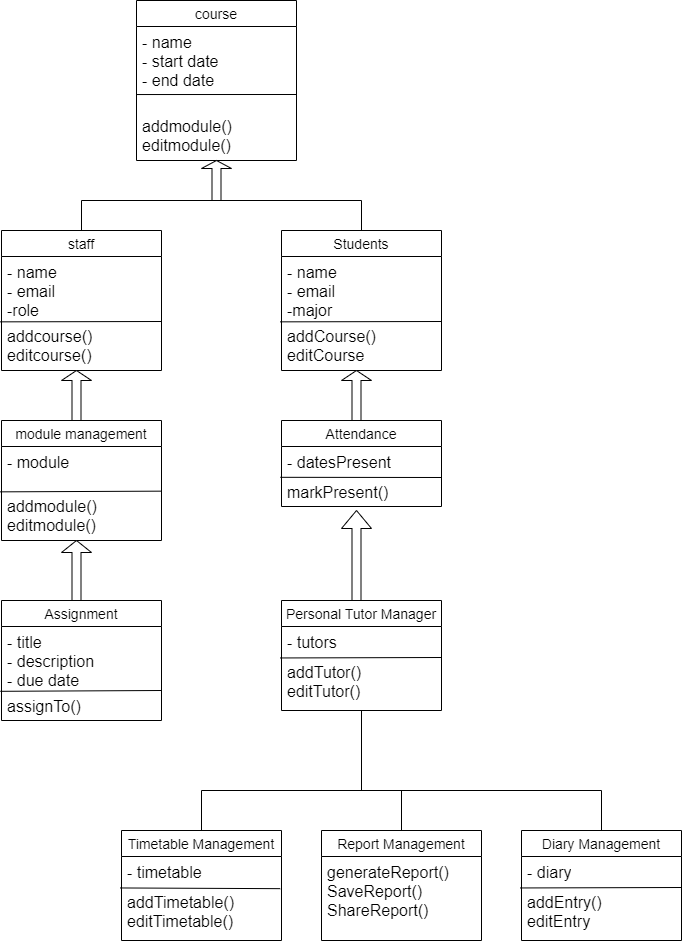
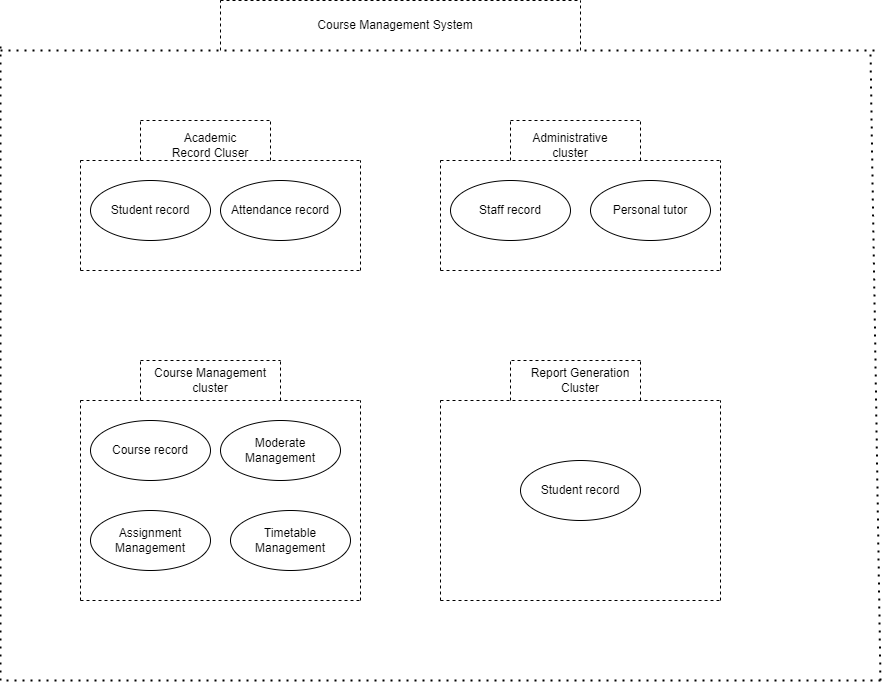


Fig : BON system chart for course management system

The BON system chart shown above offers a more comprehensive perspective on the course management system. It illustrates the various components of the system, highlights the characteristics of each component, identifies the methods or events available for interacting with the system, and describes the interconnections between the different parts of the system.

**3.2.3 BON Cluster Charts**

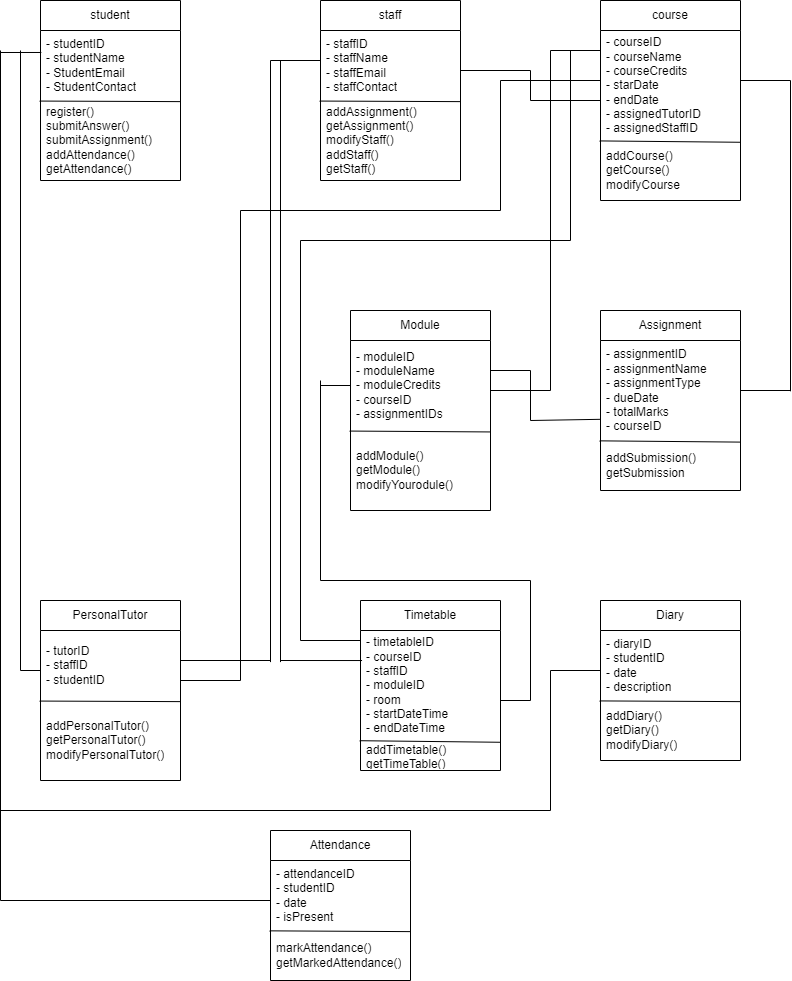
These charts show how the objects in the system are grouped into clusters based on their functions or responsibilities. This helps to identify the major functional areas of the system.



The BON cluster chart displayed above aids in recognizing the course management system's specific areas that may need extra focus, such as clusters that are highly dependent or those with a high degree of complexity. The Academic record cluster and Administration cluster have been identified as high-dependency clusters since they contain vital business objects that are necessary for the system's operation. The Course management cluster has been deemed a cluster with a high degree of complexity because it contains objects with numerous interdependencies and complicated relationships.

**3.2.4 BON Class Charts**

These charts show the detailed structure of the objects in the system, including their attributes (properties) and methods (functions). This helps to define the behavior of the system.



**3.3 Detailed Dynamic System Designs:**

**3.3.1 Events Charts**

These charts show the events that occur within the system, including their triggers, inputs, and outputs. This helps to define the system's behavior.

**3.3.2 Object Creation Charts**

These charts show how the objects in the system are created and destroyed, as well as how they interact with each other. This helps to define the lifecycle of the objects.

**3.3.3 System Scenario Charts**

These charts show how the system responds to different scenarios or use cases. This helps to test and refine the system design.

**3.3.4 Dynamic Diagrams**

These diagrams show how the system processes data and executes tasks over time. This helps to identify potential bottlenecks or performance issues.

**3.4 System Database Design:**

**3.4.1 E-R Model**

The Entity-Relationship (E-R) model is a conceptual data model that shows the entities (objects) in the system and their relationships. This helps to define the structure of the database.

**3.4.2 Attribute Listings**

These listings show the specific attributes (properties) of each entity in the system, as well as their data types and constraints. This helps to define the data schema for the database.

These are the general stages and design artifacts involved in the system analysis and design process. Depending on the specific needs and requirements of your records management system, additional stages and design artifacts may be required.