**System Analysis and Design Solution Template**

|  |  |
| --- | --- |
| **Academic year and term:** | 1 |
| **Module title:** | System Analysis and Design |
| **Module code:** | QAC020C154S |
| **Module Convener:** | Ifeoluwa Agboola |
| **Module Tutor:** | Mama Kone |
| **Student Id:** | S00062204 |
| **Deadline:** | 17/07/2020 |
| **Coursework 1:** | System Design |

|  |
| --- |
| Use case diagram |
| 1.1 Functional requirements |
| * The system shall allow the Administrators to record, delete, search and update staff, student and course related information. * The system shall allow the Tutors to record and update student attendance. * The system shall allow the feature of transferring the data from Moodle to ICIMS System. * The system shall allow the Administrators to review all grades using the ICIMS system but it should not all the Administrators to update the information. * The system shall allow the Administrators to be able to enrol students on courses and track their attendance. * The system shall allow the Tutors to record remarks on the students’ academic progress, attendance history and any other relevant information. * The system shall allow the Tutors to be able to upload lecture material, homework and exercises which the students can complete online. * The system shall notify the students automatically when their grades are ready to view. * The system shall send a warning message to students if they do not submit coursework in time and will automatically stop them from accessing the system if they do not pass three modules in one semester, and will notify admin to exclude these students from the course. * The system shall allow the Staff and students to login to the system in order to use it. |
| 1.2 use case diagram |
|  |

|  |
| --- |
| 2) use case diagram documentation |
| Primary use case 1 |
| |  |  | | --- | --- | | **Use Case Title:** | **Track Attendance** | | **Actors:** | Administrator | | **Description:** | The system shall allow the administrator to view the Attendance records of the students to Keep track. | | **Precondition** | * The administrator shall be able to access the system. * There must be some record to view for attendance. | | **Flow** | 1. The administrator access the ICIMS System. 2. The administrator navigate to attendance   Management.   1. The administrator Select a Student to Track his/her attendance. 2. The system displays the attendance record of that  particular student on the screen | |
| Primary use case 2 |
| |  |  | | --- | --- | | **Use Case Title:** | **Upload Material** | | **Actors:** | Tutor | | **Description:** | The system shall allow the tutor to upload the study material and exercise for students to download and practice. | | **Precondition** | * The tutor shall be able to access the system. * The tutor shall have an active teaching course. | | **Flow** | 1. The tutor accesses the ICIMS System. 2. The tutor navigates to the course management. 3. The tutor select a course. 4. The tutor uploads the course material. 5. The tutor sees the success message that the document is upload successfully. | |
| **10 use cases - brief description** |
| Enrol Students: The system shall allow the administrator to enrol students to different courses. |
| Student Data Maintenance: The system shall allow the administrator to manage the student records. The administrator should be able to create, update and delete record for students. |
| Exclude Notify: The system shall notify the administrator for any student that should be excluded from the course depending upon the performance of the student. |
| Track Attendance: The system shall allow the administrator to track the attendance of the students. |
| Staff Data Maintenance: The system shall allow the administrator to manage the staff records. The administrator should be able to create, update and delete record for staff. |
| Review Grades: The system shall allow the administrator to review the grades of the students. |
| Course Data Management: The system shall allow the administrator to manage the course data. The administrator should be able to create, update and delete record for Course. |
| Student Attendance Maintenance: The system shall allow the tutor to manage the student attendance. The Tutor can add, delete and update the attendance records of the students. |
| Add Remarks: The system shall allow the tutor to add remarks for any students’ performance. |
| Upload Material: The system shall allow the tutor to Upload Course Material of Exercises for students to download |
| Login: The system shall allow the students and staff to Login into the system before able to access their details. |
| View Grade: The system shall allow the students to access their grades and view their marks. |
| View Notification: The system shall allow the students to view the important notifications and announcements regarding their courses and attendance. |
| Download Material: The system shall allow the students to download the material from the system. |
| Grades Transfer: The system should have a feature to transfer grades from moodle to the local system |

|  |
| --- |
| 3) Class diagram including conceptual classes and associations, generalization, aggregation and/or composition if applicable with a brief description |
| Class Diagram |
| **Provide brief description of all key classes and main attributes:**   |  |  | | --- | --- | | Class name | Description | | Users | Users is a parent class of all the users of the ICIMS System that contains the common information and functions for all the users. This class contains all the common information Like name of user, the age of user and unique user Id. This also contains the common function among all the users that is to register to the system before able to user it and perform different operations. | | Student | Student is the Child class of the user’s class that is to represent the Students type users of ICIMS System. This class will be used to perform functions that are accessible to students. The students is needed to be logged in into the system to access it. It has the information like user id and password to login into the system. The student can perform different operations like download the course material etc. | | Teacher | Teacher is the Child class of the user’s class that is to represent the Tutor type users of ICIMS System. This class will be used to perform functions that are accessible to Tutors. The Tutors as administrator aren’t required to login into the system to use its features. The tutor can perform different operations. The teacher class is associated with other classes to access different data and to be able to perform different functions. The teacher class is associated with Student Attendance and Cloud Material Class. The teacher can make use of the Student Attendance instance to access and to manage the student attendance. Similarly the Teacher class make use of Cloud Material class instance to upload and manage the course material. The teacher can also view the details on the screen using these instances. | | Administrator | Administrator is the Child class of the user’s class that is to represent the Administrator type users of ICIMS System. This class will be used to perform functions that are accessible to Administrator. The administrator isn’t required to login into the system to use its features. The administrator can perform most of the operations of ICIMS System. The administrator class is associated with Student Record, Staff Record, Course Management and Student Attendance Classes. The administrator class make use of the Student Record the access the student records and to perform add, update and delete operations on the Student record. The administrator class is associated with the Staff Record class to perform operations on the staff record for example to update, add and delete the staff records. Moreover, the administrator class is associated with Course Management class to manage the courses and to enrol the students in different courses. Similarly, the administrator class is associated with the Student Attendance class to view and keep track of the student attendances. | | MainClass | This the class where the main control of the application is, this class is used to which is where the systems starts the operations. This class is used as the starting point to perform any function. This class is associated with the User class and using that class it will be able to perform different operations. | | CloudMaterial | This class is to maintain the course materials that will be uploaded by the teacher. The Cloud material will be accessible to the students. The students will be able to download the related course material from the system and the exercises can also be downloaded from the system. | | CourseManagement | This class is to maintain and manage the courses offered by the school. This class will be referenced by the administrator to manage such as add, update and delete the course related information. This class is associated with the Course Class that is to have the instance of the Course Object to get info for a single course and to set the related information | | Course | This class is to maintain the information related to single course such as course name, course credit hours etc. This class is associated with CourseManagement class. | | StudentRecord | This class is to maintain the data related to the students that are registered onto the system. This class is associated with the Administrator so that the administrator can access the information and can manage that information if required | | StaffRecord | This class is to maintain the data related to the staff that are registered onto the system. This class is associated with the Administrator so that the administrator can access the information and can manage that information related to staff if required | | StudentAttendance | This class is to maintain the attendance information of students that are registered and are enrolled in any of the courses. This class is associated with administrator so that administrator can keep track of the attendance of the students and can view the attendance. Also this class is associated with the Teacher so that the teacher can also record the attendance of students. | |
| 4) Interaction diagram |
| Sequence Diagram |
|  |
| Sequence Diagram - description and justifications  The sequence diagram above is for the primary use case that is creating a new course. The actor involved in this sequence diagram is the administrator. It consists of 3 Classes. One class is for the administrator that triggers the create course function. The create course function then sends a request to the Course Management class that is responsible to manage the course related information. The Course Management class then class a function of Course class by submitting the course information that is required to be created. The course class returns a course object to the Course Management class and as a result a course gets created. |
| Collaboration Diagram |
|  |
| Collaboration Diagram: description and justifications  The Collobration diagram for ICIMS System is given above. The diagram defines the interation between different objects of the System to perform a certain use case functionality. The above colloration diagram is for the Create a Course use case. The interaction starts from the MainClass that has an instance of the Users class. The User class then sends a request to the Course Management class using the Adminstrator Form. The Course Management class then interacts with the Course class that returns a course object and as a result a course gets created. |
| Flowchart/Activity Diagram |
|  |
| Flowchart Diagram Description and Justifications  The flowchart diagram of one of the primary use case of the system is given above. The above activity diagram is for enrollment of a student to a course. The flow of the diagram starts by Opening up the CIMS System by the adminstrator who is responsible to enrol a student. The next step is to navigate to the Enrollment feature of the system. The 3rd step is to select a course for which the adminstrator wants to enrol a student. The 4th step is to select a student from all the available and registered students List. In the next step the system checks if the student selected is already registered in that course or not. If the student is not registered the system enrols the student of that course and show a success message other wise the system says a error message and returns the flow of the system to the student selection section. |
| References |
| Kaur, H. and Singh, P., 2011. UML (Unified Modeling Language): standard language for software architecture development. In *International Symposium on Computing, Communication, and Control* (pp. 313-320).  Maguire, M. and Bevan, N., 2002, August. User requirements analysis. In *IFIP World Computer Congress, TC 13* (pp. 133-148). Springer, Boston, MA. |