**Project name: Course Management System**

**date: 10/08/2024**

**Team:**

**Musab Adam Ishag Etayeb**

**Mohaned Salahelidn Fadlalla**

**Chapter one**

**Project Initiation**

**System Request**

**Feasibility Analysis**

**Cost–Benefit Analysis**

**Project Selection**

**System Request**

|  |
| --- |
| **System Request**  **Course Management System Project** |
| **Project sponsor: Mohaned Salaheldin, Mosab** |
| **Business Need:** This project has been initiated to streamline the enrollment process for students and reduce administrative overhead. |
| **Business Requirements:**  By using the system, students will be able to view the available courses, and view the information related to them (credit hours, prerequisites, etc.). They should also be able to register/withdraw from courses, in addition to being able to download courses and educational modules related to their courses. The functionality that the system  should have is as follows:   * Students can search for courses by criteria like subject or instructor and view details such as schedule, prerequisites, and capacity. * Students can add courses to their schedule, with checks for prerequisites, seat availability, and time conflicts. * Students can drop or swap courses with real-time schedule updates. * Students can view and print their schedule in a calendar format. * Students receive instant notifications on enrollment status, including registrations, waitlists, and course changes. |
| **Business Value:**  The system is expected to reduce the time required for the course Enroll/Swap process by 80% (from 25 to 5 minutes), and also reduce the administrative expenses related to the registration and academic guidance office to 50%, as it will reduce the need for employees in the office from 6 to 3, and also make the study materials available 24/7.  Conservative estimates of tangible value to the company include:   * $7,200 (50% of $14,400) in annual savings from reduced administrative staff. * 80% reduction in time spent on course enrollment processes (from 25 to 5 minutes). * 24/7 access to study materials, potentially reducing the need for in-person academic guidance |
| **Special Issues or Constraints:**  The system must be fully operational and ready for use before the start of the next educational year. |
| **Steering Committee approval**  **Chair : ……………………………….… Signature:…………………………….**  **Date : ……………………………….…..** |

**Feasibility Analysis**

|  |
| --- |
| **Course Management System Analysis Executive Summary** |
| **Conducted by:MUSAB ADAM ISHAG ELTAYEB**  **Attachment:** |
| **Technical Feasibility:**  **1- Familiarity with Functional area:**  It is important to be familiar with all the characteristics and factors that contribute to the formation of a course management system, including the study materials, teachers, students and educational facilities, and how to integrate them into one system to work in the specified manner.  **2- Familiarity with Technology:**  Incorporating technological tools into the system is important and crucial to the success of the project, otherwise the system will be traditional and ineffective.  **3- Project Size:**  We aspire in our project “Course Management System” for the system to be simple and at the same time flexible, so that all users are able to deal with it in the required manner.  **4-Compatibility:**  The issue of choosing the tools and devices to create and establish the system is very important, as compatibility between the devices and technologies used must be considered. Otherwise, dealing with the system may pose some difficulties for users. |
| **Economic Feasibility:**  **1-Development costs:**  Costs are evaluated by the project manager and the project financial officer after the feasibility study.  Initially, we will set an initial amount of $2,500  **2-Annual operating costs:**  Evaluated depends on the current situation of the system and if there are any problems facing the system such as maintain or updates  **3-Annual benefits (cost savings and revenues:**  It depends on the success of the system and its effectiveness, and whether another company requests the same system with additional features.  **4-Intangible costs and benefits:**  It may be addressed and reviewed after determining the basic costs and after the project is completed. |
| **Organizational Feasibility:**  **1-Project champions:** In every project there must be leading and essential roles to lead the project, which in our project are: students, teachers and database system.  **2-Senior management :**  They are the system administrators who create, develop, and update the system (technical manager).  **3-Users:**  Users include students, teachers and sometimes third party users who are authorized to use the system for general evaluation of the system and detection of **errors.**  **4-Other stakeholders :**  General manager of educational institution.  **5-Workflow must be matched with costs during the project execution.** |
| **Additional Comments:**   * **Create initial plan for project costs to meet requirement** * **Review and update the plan constantly and continuously depends on the cycle of project execution.** * **Make requirement changes to the system as long as the phase that we achieved.** |

**Cost–Benefit Analysis**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **2023** | **2024** | **2025** | **2026** | **2027** | **Total** |
| Increased productivity value | 3000 | 3000 | 3000 |  |  | 9000 |
| Reduction in Administrative expenses | 7200 | 7200 | 7200 |  |  | 21600 |
| Reduce in Need for in-person guidance | 2000 | 2000 | 2000 |  |  | 6000 |
| **TOTAL BENEFITS:** | 12200 | 12200 | 12200 |  |  | 36600 |
| **PV OF BENEFITS: (-5%)** | **12200** | **11590** | **10980** |  |  | **34770** |
| **PV OF ALL BENEFITS:** | **12200** | **23790** | **34770** |  |  | **34770** |
| Office Space and Equipment | 1000 | 0 | 0 |  |  | 1000 |
| Software | 5000 | 0 | 0 |  |  | 5000 |
| Hardware | 2000 | 0 | 0 |  |  | 2000 |
| **TOTAL DEVELOPMENT COSTS:** | **8000** | **0** | **0** |  |  | **8000** |
| Hardware Maintenance | 1000 | 1000 | 1000 |  |  | 3000 |
| Software Maintenance | 1000 | 1000 | 1000 |  |  | 3000 |
| User training | 400 | 400 | 400 |  |  | 1200 |
| Server hosting/Cloud services | 500 | 500 | 500 |  |  | 1500 |
| **TOTAL OPERATIONAL COSTS:** | **2900** | **2900** | **2900** |  |  | **8700** |
| **TOTAL COSTS:** | **10900** | **2900** | **2900** |  |  | **15700** |
| **PV OF COSTS:** | **10900** | **2755** | **2617** |  |  | **16272** |
| **PV OF ALL COSTS:** | **10900** | **13655** | **16272** |  |  | **16272** |
| **TOTAL PROJECT BENEFITS COSTS:** | **1300** | **8835** | **8363** |  |  | **18498** |
| **YEARLY NPV:** | **1300** | **8835** | **8363** |  |  |  |
| **CUMULATIVE NPV:** | **1300** | **10135** | **18498** |  |  | **18498** |
| **RETURN ON INVESTMENT:** | **113%** | | | | | |
| **BREAK-EVEN POINT:** | **2.8 year (By the end of 2025)** | | | | | |
| **INTANGIBLE BENEFITS:** | * Better Academic Planning | | | | | |

**Project Selection**

|  |
| --- |
| **Project Selection** |
| **Size**  Q: What is the estimated duration to complete the project? A: 5 months.  Q: Is this project scalable or fixed-size project? A: fixed-size project means can be modified at any stage of project. |
| **Cost**  Q: is the project helps us to avoid high-cost?  A: it depends on the total average spent and how to organize all the costs.  Q: is it a delay of inaction will cost us too much? A: absolutely yes as long as we decrease the project efficiency and final result of the project.  Q: do we have sufficient amount of money to handle emergency situations that might  A: yes, financial executive officer of project has a plan to manage emergencies issues |
| **Purpose**  Q: What is the main goal of this project?  A: Accomplish all tasks of the Project with a required specification.  Q: What are the roles that stakeholders play in this project? A: Providing resources and tools they want in their project environment. |
| **Length**  Q: How complex is the project?  A: Medium complex project that does not need more evaluation.  Q: How do we track and measure the efficiency of our project? A: Continues Follow-up with Stakeholders to ensure they meet their requirements. |
| **Risk**  Q: What is the risk management in our project? A: Evacuation of any on of project requirement (lack of completing requirements)  Q: How do we handle conflicts between project selection criteria A: By developing decision making frame-work |
| **Scope**  Q: What is the role of project selection in achieving advantage? A: to meeting stakeholders needs  Q: Who are the targets of this project?  A: Studentes,Teachers and Managers. |
| **Return on investment**  **Q**: Is our project expected to gain revenue ?  A: Yes  Q: What’s the impact of the project on the general business? A: Assist to promote and develop the project to use it in additional features. |
| **Steering Committee approval**  **Chair : MUSAB ADAM ISHAG ELTAYEB**  **Signature:…………………………….**  **Date : 10-Aug-24** |

**ف**

**Chapter Two**

**Project Management**

**Function point estimation**

**Workplan Information**

**Gantt Evolutionary versions**

**Staffing plan\***

**CASE tools**

**Risk assessment**

**Function point estimation** **Worksheet**

**System Size estimation**

**System Components:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Description** | **Complexity** | | | |  |
| **Total Number** | **Low** | **Medium** | **High** | **Total** |
| inputs | 4 | 2\*2 | 2\*3 | 2\*6 | 22 |
| outputs | 10 | 3\*3 | 5\*4 | 2\*5 | 44 |
| Quires | 8 | 4\*2 | 2\*4 | 2\*4 | 24 |
| Files | 7 | 0\*4 | 6\*3 | 1\*1 | 19 |
| Program Interfaces | 3 | 0\*3 | 1\*2 | 2\*4 | 10 |
| **Total Unadjusted Function Points (TUFP):** | | | | | 119 |

**System Components:**

|  |  |
| --- | --- |
| Data communications | 0 |
| End-user efficiency | 1 |
| Complex processing | 0 |
| Performance | 0 |
| Distributed functions | 1 |
| Online data entry | 3 |
|  |  |
|  |  |
| **Total Processing Complexity (PC):** | 5 |

**Adjusted Processing Complexity (APC):**

**0.65 + (0.01 x 5 ) = 0.7**

**Total Adjusted Function Points (TAFP):**

**0.7 (APC) x 119 (TUFP) = 83.3 (TAFP)Required Effort Estimation**

**effort (in person-months) = 1.4 \* thousands of lines of code**

**estimated codes lines = 25,000**

**1.4 \* estimated codes lines ( 25,000) = 35 persons month of efforts.**

**Time Required Estimation**

**schedule time (months) = 3.0 \* person-months**

**3.0 \* 35 = 105**

**Workplan Information**

|  |  |
| --- | --- |
| **Task NO** | **1** |
| Name of the task | **Requirement Analyst** |
| Start date | **01-Oct-24** |
| Completion date | **14-Oct-24** |
| Person assigned to the task | **Project Manager: Mohaned salah** |
| Deliverable(s) | **Project Administration** |
| Completion status | **Open** |
| Priority | **Crucial** |
| Resources that are needed | **Requirement s** |
| Estimated time | **17 hours** |
| Actual time | **12 hours** |
|  |  |
| **Task NO** | **2** |
| Name of the task | **Establishing design** |
| Start date | **15-Oct-24** |
| Completion date | **4-Nov-24** |
| Person assigned to the task | **Project technical: Musab Adam** |
| Deliverable(s) | **IT Department** |
| Completion status | **Open** |
| Priority | **Medium** |
| Resources that are needed | **Requirements Analyst** |
| Estimated time | **10 hours** |
| Actual time | **9.5 hours** |

|  |  |
| --- | --- |
| **Task NO** | **3** |
| Name of the task | **System Development (UI)** |
| Start date | **5-Nov-24** |
| Completion date | **2-Dec-24** |
| Person assigned to the task | **Project technical: Musab Adam** |
| Deliverable(s) | **IT Department** |
| Completion status | **Open** |
| Priority | **high** |
| Resources that are needed | **Design Implementation** |
| Estimated time | **4 hours** |
| Actual time | **3 hours** |
|  |  |
| **Task NO** | **4** |
| Name of the task | **System Development (DB)** |
| Start date | **5-Nov-24** |
| Completion date | **26-Dec-24** |
| Person assigned to the task | **Project technical: Musab Adam** |
| Deliverable(s) | **IT Administration** |
| Completion status | **Open** |
| Priority | **High** |
| Resources that are needed | **Project initial Data** |
| Estimated time | **8 hours** |
| Actual time | **8 hours** |

|  |  |
| --- | --- |
| **Task NO** | **5** |
| Name of the task | **System Testing** |
| Start date | **3-Dec-24** |
| Completion date | **17-Dec-24** |
| Person assigned to the task | **Project Manager: Mohaned salah** |
| Deliverable(s) | **Project Management** |
| Completion status | **Open** |
| Priority | **Crucial** |
| Resources that are needed | **System Implementation** |
| Estimated time | **5 hours** |
| Actual time | **4.5 hours** |
|  |  |
| **Task NO** | **6** |
| Name of the task | **Documentation** |
| Start date | **18-Dec-24** |
| Completion date | **2-Jan-25** |
| Person assigned to the task | **Project Manager: Mohaned salah** |
| Deliverable(s) | **Project Management** |
| Completion status | **Open** |
| Priority | **High** |
| Resources that are needed | **Testing approval** |
| Estimated time | **4 hours** |
| Actual time | **4 hours** |

|  |  |
| --- | --- |
| **Task NO** | **7** |
| Name of the task | **System Launch** |
| Start date | **3-Jan-25** |
| Completion date | **17- Jan-25** |
| Person assigned to the task | **Project Manager: Mohaned salah** |
| Deliverable(s) | **Stakeholders** |
| Completion status | **Open** |
| Priority | **Crucial** |
| Resources that are needed | **Modules data** |
| Estimated time | **12 hours** |
| Actual time | **10 hours** |
|  |  |

**Gantt Evolutionary versions (WBS)**

| **ID** | **Task Name** | **Duration Weeks)(ً** | **Start** | **Finish** | **Predece** |  |
| --- | --- | --- | --- | --- | --- | --- |
| **1** | **Requirements Analysis** | **2** | **01/10/2024** | **14/10/2024** | **-** |  |
| **2** | **Initial Design** | **3** | **15/10/2024** | **04/11/2024** | **1** |  |
| **3** | **System Development (UI)** | **4** | **05/11/2024** | **02/12/2024** | **2** |  |
| **4** | **System Development (DB)** | **3** | **05/11/2024** | **26/12/2024** | **2** |  |
| **5** | **System Testing** | **2** | **03/12/2024** | **17/12/2024** | **3, 4** |  |
| **6** | **Documentation** | **2** | **18/12/2024** | **02/01/2025** | **5** |  |
| **7** | **System Launch** | **1** | **03/01/2025** | **17/01/2025** | **6** |  |

**Staffing plan\***

|  |  |  |  |
| --- | --- | --- | --- |
| **Role** | **Description** | **Assigned To**  **All the team\*** | **Manger**  **At this phase\*** |
| **Business Analyst** | Responsible for gathering and analyzing project requirements from end-users and ensuring the system aligns with business objectives |  | **Mohaned** |
| **Programmer** | Writing and implementing code to develop the course management system, ensuring technical performance and efficiency |  | **Mohaned** |
| **Quality Assurance** | Testing the system and ensuring quality before launch, verifying that all user requirements are met |  | **MUSAB** |
| **Technical Lead** | Managing the technical aspects of the project and guiding the development team in technological decisions |  | **MUSAB** |
| **Project Manager** | Responsible for planning and overseeing all aspects of the project, ensuring progress according to the timeline and budget |  | **Mohaned** |
|  | | | |

**At this phase\*: Select a manger from team for each phase depend on his/her qualifications.**

**CASE tools**

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Name** | **Use at Phase** | **Using reason** |
|  | Requirements Gathering Management | upper CASE | The initial stage of collecting basic information to create a database |
|  | Modeling | upper CASE | Used for representation such as designing diagrams and data , process modling |
|  | Code Generation | lower CASE | Automatic generate codes to minimize time taken for manual code,also for ensure accuracy of the code |
|  | Configuration Management | lower CASE | Helps for changes codes and can helps for track and prevent risks |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Risk assessment**

|  |  |
| --- | --- |
| **RISK 1** | Some team members may not be familiar with the technology or programming language used, such as Dart (for Flutter development) or Java. |
| **Likelihood of risk:** | High |
| **Potential impact on the project:** | This can lead to significant delays in development by 50%. |
| **Ways to address this risk:**  Provide training sessions for developers before starting the project. Additionally, consider bringing in an external expert to guide the team and reduce the learning curve. | |
| **RISK 2** | The system may have difficulties integrating with existing university systems, such as databases or authentication mechanisms. |
| **Likelihood of risk:** | Medium |
| **Potential impact on the project:** | Failure to integrate properly could limit the functionality of the system, such as accessing student records or course schedules. |
| **Ways to address this risk:**  Perform thorough testing and engage with IT personnel at the university early in the project to ensure compatibility. | |
| **RISK 3** | The system will handle sensitive student information, making it a target for security breaches. |
| **Likelihood of risk:** | High |
| **Potential impact on the project:** | A data breach could result in legal consequences and loss of trust. |
| **Ways to address this risk:**  Implement strong encryption, authentication, and security protocols. Regularly conduct security audits and patches. | |
|  | |

|  |  |
| --- | --- |
| **RISK 4** | Users, such as students and faculty, may resist adopting the new system or may not be adequately trained to use it. |
| **Likelihood of risk:** | Medium |
| **Potential impact on the project:** | Lack of user adoption could render the system underutilized, reducing its value. |
| **Ways to address this risk:**  Conduct training sessions, offer user-friendly documentation, and gather feedback during development to ensure the system is intuitive. | |

**Chapter Three**

**Requirements Determination**

**Requirements Analysis Strategies**

**Requirements-Gathering Techniques**

**Requirements Definition**

**System Proposal**

**Iterative Plan**

**Requirements Analysis Strategies**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Business Process Reengineering** | | | | | |
| **☑** | **Outcome Analysis** | **□** | **Technology Analysis** | **□** | **Activity Elimination** |
| **Outcome Analysis**  The purpose of the outcome analysis is to identify the ultimate goals and expectations that the university aims to achieve through the implementation of the new Course Management System. Under the current strict academic calendar system, the educational process lacks the flexibility that students need to achieve the best academic experience. Through business process reengineering, we aim to achieve the following outcomes:   1. **Increased Student Autonomy:** Students will have the ability to select courses that align with their academic and professional goals, which will enhance their motivation and engagement in learning. 2. **Improved Access to Information**: The new system will provide comprehensive information about course content, prerequisites, and schedules, making it easier for students to make informed decisions about their academic paths. 3. **Enhanced User Experience:** Students will be able to enroll in courses online, saving them time and effort, and they will receive instant notifications regarding any changes to course schedules. 4. **Provision of Academic Support:** By improving administrative efficiency and reducing the burden on staff, the administration team will be able to focus their efforts on providing academic support to students, contributing to improved academic performance and increased student satisfaction.   Based on this analysis, the new system will effectively meet students' needs, enhancing the quality of education provided and helping the university achieve its strategic objectives. | | | | | |

**Requirements-Gathering Techniques**

**A. Interview**

**Interview Schedule**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Position** | **Purpose of Interview** | **Meeting** |
| Dr. Waffaa | Dean | Gather insights on the Course Management System and its requirements | 1-Oct-24 |

**Interview Questions**

|  |
| --- |
| **Closed-ended questions** |
| **Q:**Do you believe the current course management system adequately meets the needs of students and faculty? |
| **A:**No |
|  |

|  |
| --- |
| **Open-ended questions** |
| **Q:** What challenges do you see with the current course management system? |
| **A**: The current system lacks flexibility, making it difficult for students to manage their courses effectively. |
|  |
| **Q**: How do these challenges affect student engagement and performance? |
| **A**: Students often feel overwhelmed by the lack of choice, which can lead to disengagement and affect their academic success. |

|  |
| --- |
| **Probing questions** |
| **Q**: Can you elaborate on how faculty members are impacted by the current system? |
| **A**: Faculty spend excessive time assisting students with enrollment-related inquiries, detracting from their teaching responsibilities. |
|  |
| **Q**: What specific features do you think would improve the Course Management System? |
| **A**: A course recommendation feature based on student interests and a robust notification system for deadlines would be beneficial |

**Interview report**

|  |
| --- |
| **Interview Notes Approved by: Mohaned Salah** |
| **Person Interviewed:** Dr. Waffaa |
| **Interviewer: Mohaned** |
| **Purpose of Interview:**   * Gather requirements for the Course Management System |
| **Summary of Interview:**   * Discussed current enrollment challenges and potential improvements. * Explored ideas for enhancing student engagement and administrative efficiency. * Dr. Waffaa emphasized the need for a user-friendly interface and real-time updates for students. |
| **Open Items:**   * Clarification needed on specific functionalities desired. * Additional input from faculty regarding course selection. |
| **Detailed Notes:**  **Current Challenges:**   * Dr. Waffaa pointed out that the existing system does not allow for any flexibility in course selection, which can lead to dissatisfaction among students. * The strict academic calendar means all students take the same courses, which doesn’t cater to individual learning needs or interests.   **Impact on Students:**   * Students often feel overwhelmed by the lack of choice and may disengage from their studies. * The inability to adjust course loads can affect their academic performance and overall satisfaction with the educational experience.   **Faculty Concerns:**   * Faculty members spend excessive time assisting students with enrollment-related inquiries, detracting from their teaching time. * There is a need for improved communication between students and faculty regarding course offerings and requirements.   **Desired Features:**   * Dr. Waffaa suggested implementing a course recommendation feature based on student interests and academic progress. * A robust notification system is essential to alert students about important deadlines, such as registration and course changes.   **Suggestions for Improvement:**   * The new system should provide a dashboard for students to view their schedules, course information, and alerts in one place. * Integrating a feedback mechanism would allow students to express their concerns and suggestions about the courses they are enrolled in. |

1. **Questionnaire**
2. **Faculty Members Questionnaire**

**Questionnaire Schedule**

|  |  |
| --- | --- |
| **Participants area** | Faculty Members (Full-time, Part-time, Adjunct professors) |
| **Number of samples** | 10 |
| **Website link (if its online)** | - |
| **Measurement to validity** | * **Cronbach's Alpha** to assess the reliability of the questionnaire. * Descriptive **statistics (mean, median, mode)** to analyze response trends. * **Chi-square tests or ANOVA** to determine if there are significant differences between groups (e.g., full-time vs part-time faculty). * **Correlation analysis** to identify relationships between system usage and satisfaction levels. |
| **approval committee** | * Dr. Waffaa, Head of the Faculty of Computer Science * Institutional Review Board (IRB) of SUST * Academic Affairs Committee |
| **Deadline** | 15/10/2024 |

**Questionnaire Questions**

| **Question** | **Answers** |
| --- | --- |
| **What is your primary role at the university?** | ( ) Full-time faculty  ( ) Part-time faculty  ( ) Adjunct professor |
| **How many years have you been teaching at this university?** | ( ) 1-3 years  ( ) 4-6 years  ( ) 7+ years |
| **Do you currently interact with the course management system for enrolling students or managing courses?** | ( ) Yes  ( ) No |
| **How often do you assist students with course registration issues during the enrollment period?** | ( ) Frequently  ( ) Occasionally  ( ) Rarely |
| **What are the most common challenges you face when interacting with the current course management system? (Select all that apply)** | ( ) Lack of real-time updates.  ( ) Difficulty in managing course prerequisites and requirements.  ( ) Time-consuming administrative tasks.  ( ) Communication issues with students regarding enrollment |
| **Which features would you like to see improved or added in the new system? (Select all that apply)** | ( ) Real-time student enrollment notifications  ( ) Course capacity management  ( ) A better system for tracking student progress  ( ) Simplified course prerequisites checking  ( ) Other (please specify) \_\_\_\_\_\_\_ |
| **How important is it to have a course recommendation system for students based on their academic progress?** | ( ) Very important  ( ) Important  ( ) Neutral  ( ) Not important |
|  |  |

**Questionnaire Follow-up**

|  |  |
| --- | --- |
| **Analysis technique** | Statistical analysis (e.g., frequency analysis, cross-tabulation). |
| **Results** | Will be based on the collected data. |

1. **Students Questionnaire**

**Questionnaire Schedule**

|  |  |
| --- | --- |
| **Participants area** | Students (Freshman, Sophomore, Junior, Senior) |
| **Number of samples** | 20 |
| **Website link (if its online)** | - |
| **Measurement to validity** | * **Cronbach's Alpha** to assess the reliability of the questionnaire. * Descriptive **statistics (mean, median, mode)** to analyze response trends. * **Chi-square tests or ANOVA** to determine if there are significant differences between groups (e.g., full-time vs part-time faculty). * **Correlation analysis** to identify relationships between system usage and satisfaction levels. |
| **approval committee** | * Dr. Waffaa, Head of the Faculty of Computer Science * Institutional Review Board (IRB) of SUST * Academic Affairs Committee |
| **Deadline** | 15/10/2024 |

**Questionnaire Questions**

| **Question** | **Answers** |
| --- | --- |
| **What is your current academic year?** | ( ) Freshman  ( ) Sophomore  ( ) Junior  ( ) Senior |
| **How often do you use the current system for course enrollment?** | ( ) Frequently  ( ) Occasionally  ( ) Rarely |
| **How would you rate your experience with the current course management system?** | ( ) Very satisfied  ( ) Satisfied  ( ) Neutral  ( ) Dissatisfied  ( ) Very dissatisfied |
| **What features would you like to see in the new system? (Select all that apply)** | ( ) Course search by criteria (e.g., subject, instructor)  ( ) Real-time updates on seat availability  ( ) Schedule conflict checking ( ) Easy course drop/swap process  ( ) Instant notifications for enrollment status |
| **What are the most common challenges you face when interacting with the current course management system? (Select all that apply)** | ( ) Lack of real-time updates.  ( ) Difficulty in managing course prerequisites and requirements.  ( ) Time-consuming administrative tasks.  ( ) Communication issues with students regarding enrollment |
| **How important is it to have 24/7 access to course materials and schedules?** | ( ) Very important  ( ) Important  ( ) Neutral  ( ) Not important |
|  |  |

**Questionnaire Follow-up**

|  |  |
| --- | --- |
| **Analysis technique** | Statistical analysis (e.g., frequency analysis, cross-tabulation). |
| **Results** | Based on gathered feedback. |

**Document Analysis**

**Attach documents with comments as images**

**Requirements Definition**

|  |
| --- |
| **Nonfunctional Requirements:**  **1. Operational Requirements**  1.1. The system will operate in Andriod,IOS,Windows and Macintosh environments.  1.2. The system will be able to receive inputs form users .  1.3. The system will be able to upload respective modules chapters to students who selected relevant modules.  **2. Performance Requirements**  2.1 should be adaptable with all operating systems.  **3. Security Requirements**  3.1. Containing at least Minimum-security software to ensure confidentiality, Integrity and Availability.  **4. Cultural and Political Requirements**  4.1. No special cultural and political requirements are anticipated |
| **Functional Requirements**  Courses:  -student should select which selective modules he can choose  -student can determine which intake to start enrolling.  -student can ask for changing specialism.  -student should apply for online survey after each semester completion.  Online Platform:  -this platform used for submission of assignment  - a teacher can upload assignments guidelines and procedures for submit.  -a student can submit, view and review assignment status |

**System Proposal**

|  |
| --- |
| **1. Table of Contents** |
| **2. Executive Summary**  Implementing a Course Management System is a strategic investment that will improve the efficiency and effectiveness of course administration. By automating processes, enhancing communication, and providing a more personalized learning experience, the CMS will benefit both students and faculty, ultimately leading to improved academic outcomes. |
| **3. System Request**  **Business Need: Business Need:** This project has been initiated to streamline the enrollment process for students and reduce administrative overhead  **Business Requirements:**  By using the system, students will be able to view the available courses, and view the information related to them (credit hours, prerequisites, etc.). They should also be able to register/withdraw from courses, in addition to being able to download courses and educational modules related to their courses. The functionality that the system  should have is as follows:  **Business Value:**  The system is expected to reduce the time required for the course Enroll/Swap process by 80% (from 25 to 5 minutes), and also reduce the administrative expenses related to the registration and academic guidance office to 50%, as it will reduce the need for employees in the office from 6 to 3, and also make the study materials available 24/7.  Conservative estimates of tangible value to the company include:  **Special Issues or Constraints:**  The system must be fully operational and ready for use before the start of the next educational year. |
| **4. Workplan**  there are 7 essential tasks in the system ,ranged between high and medium priority, which are starts with Requirement Analysis to identify all requirements ,secondly creating a design diagram ,thirdly development a system with user interface ,fourthly developing a system database ,fifthly starting testing whole system , sixthly a Documentation of a system , finally Launch a system |
| **5. Feasibility Analysis**  Feasibility analysis categories :  1-Technical Feasibility which contains Familiarity with Functional area, Technology ,Project Size and Compatibility  2-Economic Feasibility contains Development costs,Annual operating costs,Annual benefits and Intangible costs and benefits.  3-Organizational Feasibility contains Project champions, Senior management ,Users and Other stakeholders and Workflow must be matched with costs during the project execution. |
| **6. Requirements Definition**  **Nonfunctional Requirements:**  **1. Operational Requirements**  1.1. The system will operate in Andriod,IOS,Windows and Macintosh environments.  1.2. The system will be able to receive inputs form users .  1.3. The system will be able to upload respective modules chapters to students who selected relevant modules.  **2. Performance Requirements**  2.1 should be adaptable with all operating systems.  **3. Security Requirements**  3.1. Containing at least Minimum-security software to ensure confidentiality, Integrity and Availability.  **4. Cultural and Political Requirements**  4.1. No special cultural and political requirements are anticipated  **Functional Requirements**  Courses:  -student should select which selective modules he canchoose  -student can determine which intake to start enrolling.  -student can ask for changing specialism.  -student should apply for online survey after each semester completion.  **Online Platform:**  -this platform used for submission of assignment  - a teacher can upload assignments guidelines and procedures for submit.  -a student can submit, view and review assignment status |
| **7. Functional Model**  -will be provided in Chapter 4 |
| **8. Structural Models**  - will be provided in Chapter 5 |
| **9. Behavioral Models**  - will be provided in Chapter 6 |
| **Appendices**   * Discussed current enrollment challenges and potential improvements. * Explored ideas for enhancing student engagement and administrative efficiency. * Dr. Waffaa emphasized the need for a user-friendly interface and real-time updates for students. |

**Chapter Four**

**Functional Modeling**

**Activity Diagrams**

**Use-Case**

**Use-Case Point–Estimation**

**Activity Diagrams**

|  |  |
| --- | --- |
| **Activity Diagram (1)** | **Name: System Login Process** |
| System Login Process | |
| **Description** | |
| This diagram describes the process of logging into the system. It begins when the user (student or admin) enters their credentials, which the system validates. If the credentials are valid, the user is logged in, and their respective dashboard (home page) is displayed. If not, the user is prompted to retry or reset their password. | |

|  |  |
| --- | --- |
| **Activity Diagram (2)** | **Name: Course Enrollment Process** |
| Course Enrollment Process | |
| **Description** | |
| This diagram represents the workflow of a student enrolling in a course. The process starts when the student searches for available courses, selects one, and the system checks seat availability. If seats are available, the student enrolls, and the system updates their schedule and confirms the enrollment. If the course is full, the student must search for another course. | |

|  |  |
| --- | --- |
| **Activity Diagram (3)** | **Name: Course Swap Process** |
| Course Swap Process | |
| **Description** | |
| The course swap process allows a student to replace a currently enrolled course with another. The student selects the course to be swapped, searches for a new course, and the system checks if there are seats available. If the new course has seats, the student confirms the swap, and the system updates the schedule and sends confirmation. | |

|  |  |
| --- | --- |
| **Activity Diagram (4)** | **Name: Notifications for Course Changes** |
| Notifications for Course Changes | |
| **Description** | |
| This diagram illustrates how the system sends notifications to students when there are changes to their enrolled courses. The system detects updates (e.g., time changes or cancellations), checks if students are enrolled, generates notifications, and sends them via the chosen medium (email, SMS, or in-app). | |

|  |  |
| --- | --- |
| **Activity Diagram (5)** | **Name: Administering Course Data** |
| Administering Course Data | |
| **Description** | |
| This diagram covers the process where an admin manages courses by adding, updating, or deleting them. The admin selects the appropriate action (add, update, delete), provides or modifies course details, and the system validates and applies the changes. The system confirms the updates, and notifications may be sent to affected students if necessary. | |

**Use-Case Description**

|  |  |  |  |
| --- | --- | --- | --- |
| **Description** | | | |
| **Use-Case Name: Course Management System** | **ID: 1** | | **Importance Level: High** |
| **Primary Actor: Student** | | **Use Case Type: Detail, essential** | |
| **Stakeholders and Interests:**  1-Student  2-Teacher  3-Management  4-IT Staff | | | |
| **Brief Description: This use case describes how a student can select and manages his/her modules on each semester as well as accessing courses and upload assignments on platform.** | | | |
| **Trigger: a student chooses a limited number of modules each study year.**  **Type:** **External** | | | |
| **Relationships:**  **Association: Student**  **Include: select modules arrangement**  **Extend: upload assignments**  **Generalization:** | | | |
| **Normal Flow of Events:**   1. Student starts his first year 2. Student pays fee to management 3. Student he/she has choice number of course on each semester. 4. Teacher upload material to online platform 5. Student download course materials from online platform 6. Student upload assignments to online platform for submission 7. Student can view his/her assessment. 8. IT staff and Management can edit ,update and control database and System. | | | |
| **Subflows:**  **S-1 New Semester:**   1. Management asks students to select modules for each semester   **S-2 Change Module:**   1. Students submit form for changing an existing module 2. Management verifies credit hours of new module selected by student | | | |
| **Alternate/Exceptional Flows:**   1. The student executes the Create select modules use case. 2. Student can change his course | | | |

**Use-Case Diagram**

**A diagram of a course management system

Description automatically generated**

**Use-Case Point–Estimation**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Unadjusted Actor Weighting Table:** | | | | |
| **Actor Type** | **Description** | **Weighting Factor** | **Number** | **Result** |
| Simple | **External System using a protocol-based** | **1** | **0** | **0** |
| Average | **External System using a protocol-based** | **2** | **0** | **0** |
| Complex | **Human** | **4** | **4** | **16** |
| **Unadjusted Actor Weight Total (UAW)** | | | | **16** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Unadjusted Use Case Weighting Table:** | | | | |
| **Use-CaseType** | **Description** | **Weighting Factor** | **Number** | **Result** |
| Simple | **1–2 transactions** | **4** | **1** | **4** |
| Average | **3–5 transactions** | **8** | **4** | **32** |
| Complex | **>5 transactions** | **13** | **0** | **0** |
| **Unadjusted Use Case Weight Total (UUCW)** | | | | **36** |

|  |
| --- |
| **Unadjusted use-case points (UUCP) = UAW + UUCW = 16 + 36 = 52** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Factor/s Complexity Factors:** | | | | | |
| **Factor Number** | **Description** | **Weight** | **Assigned Value (0–5)** | **Weighted Value** | **Notes** |
| **T1** | **Distributed system** | **2.0** | **0** | **0** |  |
| **T2** | **End-user online efficiency** | **1.0** | **3** | **3** |  |
| **T3** | **Easy to install** | **1.0** | **1** | **1** |  |
| **T4** | **Ease of use** | **1.0** | **4** | **2** |  |
| **T5** | **Ease of change** | **1.5** | **2** | **3** |  |
| **T6** | **Special security objectives included** | **1.0** | **0** | **1** |  |
| **X Factor Value (X first letter Factor)** | | | | **10** |  |

**calculate**

|  |  |  |
| --- | --- | --- |
| **Calculations Results** | | |
| **Technical Complexity Factor (TCF) = 0.6 (0.01\*TFactor)** | **0.6 + (0.01 \* 10)** | **0.7** |
| **Adjusted use case points (UCP) = UUCP\*TCF\*ECF** | **52 \* 0.7** | **36.4** |
| **Effort in person-hours = UCP\*PHM** | **36.4\*20** | **728** |

**Iterative Plan**

plan steps are reexamined and revised, and another decision is made by the project sponsor and approval committee about whether to terminate the project or continue.

**Chapter Four**

**Structural Modeling**

**CRC Cards**

**Class Diagram**

**Object Diagrams**

**Object Identification**

**Model Reviewing**

**CRC Cards**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Front:** | | | | | | | | | |
|  | | **Class Name:**Student | | **ID:**C001 | | | | **Type:**Concrete, Domain |  |
|  | | **Description:**A user who can enroll in courses, view notifications, and manage their schedule. | | | | | | **Associated Use Cases:** |  |
|  | | **Responsibilities** | | | | **Collaborators** | | |  |
|  | |  | | | |  | | |  |
|  | | Enroll in courses | | | |  | | |  |
|  | | Swap courses | | | | Course | | |  |
|  | | View course schedule | | | | Enrollment Manager | | |  |
|  | | Receive notifications for course updates | | | | Notification Manager | | |  |
|  | |  | | | |  | | |  |
| **Back:** | | | | | | | | | |
|  | **Attributes:** | | | | | | | | |
|  | |  | | |  | |  | |  |
|  | | Student ID (text) | | |  | |  | |  |
|  | | Enrolled courses (list of course IDs) | | |  | |  | |  |
|  | | Notifications (list of messages) | | |  | |  | |  |
|  | | Schedule (list of times and courses) | | |  | |  | |  |
|  | |  | | |  | |  | |  |
|  | **Relationships:** | | | | | | | | |
|  | | | **Generalization (a-kind-of):** | User | | | | |  |
|  | | |  | | | | |  |
|  | | | **Aggregation (has-parts):** |  | | | | |  |
|  | | |  | | | | |  |
|  | | | **Other Associations:** | Course,Enrollment Manager,Notification Manager | | | | |  |
|  | | |  | | | | |  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Front:** | | | | | | | | | |
|  | | **Class Name:**Teacher | | **ID:**C002 | | | | **Type:** |  |
|  | | **Description:**A user who manages courses and interacts with students enrolled in those courses. | | | | | | **Associated Use Cases:** · Log in  · View assigned courses  · Update course details  · Send notifications |  |
|  | | **Responsibilities** | | | | **Collaborators** | | |  |
|  | |  | | | |  | | |  |
|  | | Update course details | | | | Course | | |  |
|  | | Notify students of updates | | | | Notification Manager | | |  |
|  | | View list of enrolled students | | | | Student | | |  |
|  | |  | | | |  | | |  |
| **Back:** | | | | | | | | | |
|  | **Attributes:** | | | | | | | | |
|  | |  | | |  | |  | |  |
|  | | Teacher ID (text) | | |  | |  | |  |
|  | | Assigned courses (list of course IDs) | | |  | |  | |  |
|  | | Enrolled students (list of student IDs) | | |  | |  | |  |
|  | |  | | |  | |  | |  |
|  | **Relationships:** | | | | | | | | |
|  | | | **Generalization (a-kind-of):** | User | | | | |  |
|  | | |  | | | | |  |
|  | | **Aggregation (has-parts):** | |  | | | | |  |
|  | |  | | | | |  |
|  | | **Other Associations:** | |  | | | | |  |
|  | | Course, Student, Notification Manager | | | | |  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Front:** | | | | | | | | | |
|  | | **Class Name:**Admin | | **ID:**C003 | | | | **Type:** Concrete, Domain |  |
|  | | **Description:**A system administrator who manages data, configurations, and system settings. | | | | | | **Associated Use Cases:**· Manage course data  · Manage user accounts (students and teachers)  · View system reports |  |
|  | | **Responsibilities** | | | | **Collaborators** | | |  |
|  | |  | | | |  | | |  |
|  | | Manage courses | | | | Course | | |  |
|  | | Manage user data (students, teachers) | | | | Student | | |  |
|  | | Configure system settings | | | | Teacher | | |  |
|  | | View system reports | | | |  | | |  |
|  | |  | | | |  | | |  |
| **Back:** | | | | | | | | | |
|  | **Attributes:** | | | | | | | | |
|  | | Admin ID (text) | | |  | |  | |  |
|  | | Courses managed (list of course IDs) | | |  | |  | |  |
|  | | User data (list of student and teacher IDs) | | |  | |  | |  |
|  | |  | | |  | |  | |  |
|  | **Relationships:** | | | | | | | | |
|  | | | **Generalization (a-kind-of):** |  | | | | |  |
|  | | | User | | | | |  |
|  | | | **Aggregation (has-parts):** |  | | | | |  |
|  | | |  | | | | |  |
|  | | | **Other Associations:** | Associations: Course, Student, Teacher | | | | |  |
|  | | |  | | | | |  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Front:** | | | | | | | | | |
|  | | **Class Name:**Course | | **ID:**C004 | | | | **Type:**Concrete, Domain |  |
|  | | **Description:**Represents a course that students enroll in and teachers manage. | | | | | | **Associated Use Cases:**· Store course information  · Track enrolled students and assigned teachers |  |
|  | | **Responsibilities** | | | | **Collaborators** | | |  |
|  | |  | | | |  | | |  |
|  | | Track student enrollments | | | | Student | | |  |
|  | | Track course seat availability | | | | Teacher | | |  |
|  | | Store course details (e.g., schedule, description) | | | | Enrollment Manager | | |  |
|  | |  | | | |  | | |  |
| **Back:** | | | | | | | | | |
|  | **Attributes:** | | | | | | | | |
|  | | Course name (text) | | |  | |  | |  |
|  | | Course description (text) | | |  | |  | |  |
|  | | Schedule (list of times) | | |  | |  | |  |
|  | | Enrolled students (list of student IDs) | | |  | |  | |  |
|  | | Assigned teacher (teacher ID) | | |  | |  | |  |
|  | |  | | |  | |  | |  |
|  | **Relationships:** | | | | | | | | |
|  | | | **Generalization (a-kind-of):** | EducationalEntity | | | | |  |
|  | | |  | | | | |  |
|  | | | **Aggregation (has-parts):** |  | | | | |  |
|  | | |  | | | | |  |
|  | | | **Other Associations:** | Student, Teacher, Enrollment Manager | | | | |  |
|  | | |  | | | | |  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Front:** | | | | | | | | | |
|  | | **Class Name:**Enrollment Manager | | **ID:**C005 | | | | **Type:**Process |  |
|  | | **Description:**Manages course enrollments, swaps, and drops for students. | | | | | | **Associated Use Cases:**· Enroll students in courses  · Swap or drop courses |  |
|  | | **Responsibilities** | | | | **Collaborators** | | |  |
|  | |  | | | |  | | |  |
|  | | Manage course enrollment and availability | | | | Student | | |  |
|  | | Handle course swaps and drops | | | | Course | | |  |
|  | |  | | | |  | | |  |
| **Back:** | | | | | | | | | |
|  | **Attributes:** | | | | | | | | |
|  | |  | | |  | |  | |  |
|  | | Enrollment data (list of enrollments) | | |  | |  | |  |
|  | | Course availability (number) | | |  | |  | |  |
|  | |  | | |  | |  | |  |
|  | **Relationships:** | | | | | | | | |
|  | | | **Generalization (a-kind-of):** |  | | | | |  |
|  | | |  | | | | |  |
|  | | | **Aggregation (has-parts):** |  | | | | |  |
|  | | |  | | | | |  |
|  | | | **Other Associations:** |  | | | | |  |
|  | | | Student, Course | | | | |  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Front:** | | | | | | | | | |
|  | | **Class Name:**Notification Manager | | **ID:**C006 | | | | **Type:**Process |  |
|  | | **Description:**Sends notifications to students and teachers about course-related updates. | | | | | | **Associated Use Cases:**· Send notifications for course changes  · Alert for class updates or cancellations |  |
|  | | **Responsibilities** | | | | **Collaborators** | | |  |
|  | |  | | | |  | | |  |
|  | | Generate and send notifications | | | | Student | | |  |
|  | | Track notification history | | | | Teacher | | |  |
|  | |  | | | |  | | |  |
| **Back:** | | | | | | | | | |
|  | **Attributes:** | | | | | | | | |
|  | |  | | |  | |  | |  |
|  | | Notification data (list of notifications) | | |  | |  | |  |
|  | |  | | |  | |  | |  |
|  | **Relationships:** | | | | | | | | |
|  | | | **Generalization (a-kind-of):** |  | | | | |  |
|  | | |  | | | | |  |
|  | | | **Aggregation (has-parts):** |  | | | | |  |
|  | | |  | | | | |  |
|  | | | **Other Associations:** | Student, Teacher | | | | |  |
|  | | |  | | | | |  |

**Class Diagram**

|  |
| --- |
| **UML class** |

**Object Diagrams**

|  |
| --- |
| 1. **Student Enrollment Object Diagram**   **1**   1. **Teacher-Student Interaction Object Diagram**   **2**   1. **Admin Managing Courses Object Diagram**   **3**   1. **Notification Object Diagram**   **4** |

**Object Identification**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **□** | **Textual Analysis** | **□** | **Common Object List** | **□** | **Patterns** |
| **□Textual Analysis:**   1. Student starts his first year 2. Student pays fee to management 3. Student he/she has choice number of course on each semester. 4. Teacher upload material to online platform 5. Student download course materials from online platform 6. Student upload assignments to online platform for submission 7. Student can view his/her assessment.   **□Common Object List:**  Physical and tangible things:   1. Books 2. Libraries 3. Offices 4. Equipment 5. Resources   **□Patterns:**  Many different types of patterns have been proposed, ranging from high-level business-oriented patterns to more low-level design patterns   |  |  | | --- | --- | | Business Domain | Sources of Patterns | | Student Enrollment | 3,4 | | Teacher-Student Interaction | 3 | | Admin Managing Courses | 2 | | Notifications | 1 | | | | | | |

|  |
| --- |
| **Class Diagram Model Review:**   1. **User:**   Each user has a particular Id and name to sing into system.  **2-After Signing in done, process divided into three stages:**  **2.1Student:**  2.1.1: each student has a unique ID.  2.1.2: students can choose courses based on lists of courses.  2.1.3: students have a choice to receive notifications.  2.1.4: review Schedules.  **2.2Teacher:**  2.2.1: Each teacher has unique ID.  2.2.2: Teachers issue courses and modules materials.  2.2.3: Teachers enrolled students who are eligible for a specified course.  2.2.4: Teachers generate their own schedule and pass it to Administration.  **2.3Admin:**  2.3.1: each admin has a unique ID.  2.3.2: admin can view, adjust, delete and manage courses.  2.3.3: admin receive and process data from all users which were entered through system input functions.  2.3.4: admin can schedule and reschedule intakes and courses. |

**Chapter Five**

**Behavioral Modeling**

**Sequence Diagrams**

**Communication Diagrams**

**State Machine Diagrams**

**CRUD Analysis**

**Sequence Diagrams**

|  |
| --- |
| **A diagram of a computer program  Description automatically generated** |

**Communication Diagrams**

|  |
| --- |
| **A diagram of a diagram  Description automatically generated**  A diagram of a course  Description automatically generated |

**State Machine Diagrams**

|  |
| --- |
| 1. **Student Enrollment State Machine Diagram**   **10**   1. **Course State Machine Diagram**   **20**   1. **Notification State Machine Diagram**   **30**   1. **Teacher Interaction State Machine Diagram**   **40** |

**CRUD Analysis**

**CRUD Matrix**

| **Classes** | **Student (Actor)** | **Teacher (Actor)** | **Admin (Actor)** | **Course** | **Enrollment** | **Notification** | **Schedule** | **Assignment** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Student** | - | - | C, R, U, D | R | C, R, U, D | R | R | R |
| **Teacher** | R | - | C, R, U, D | R | - | R | R | C, R, U, D |
| **Admin** | - | R | - | R, U | - | C, U, D | C, U, D | - |
| **Course** | R | C, U, D | C, U, D | - | - | - | - | R |
| **Enrollment** | C, R, U, D | R | C, U, D | C | - | - | - | - |
| **Notification** | R | R | C, R, U, D | - | - | - | - | - |
| **Schedule** | R | R, U | C, R, U, D | - | - | - | - | - |
| **Assignment** | R | C, U, D | C, U, D | - | - | - | - | - |

**CRUD Analysis**

1. **Student Interactions:**

* Student can create, read, update, and delete their own enrollments, interact with courses, and receive notifications.
* They can only read the teacher, course schedules, and assignments.

1. **Teacher Interactions:**

* Teacher can create, update, and delete assignments and course schedules.
* They have read access to students, enrollments, and notifications.

1. **Admin Interactions:**

* Admin has the most privileges. They can create, read, update, and delete students, teachers, courses, notifications, and schedules.
* They oversee all interactions in the system and manage everything from the backend.

1. **Course, Enrollment, and Notifications:**

* These entities are mainly interacted with by students, teachers, and admins, performing a range of CRUD operations based on their role in the system.

**Project name:**

**date:**

**Chapter Six**

**Moving on to Design**

**Package Diagram**

**Package Diagram**

|  |
| --- |
| **A screenshot of a computer  Description automatically generated** |

**Chapter Seven**

**Class and Method Design**

**Revised Class Diagram**

**Revised CRC Cards**

**Method Specification Forms**

**Method Contracts**

**Revised Class Diagram**

|  |
| --- |
| **Class Diagram.drawio** |

**Revised CRC Cards**

|  |
| --- |
| **CRC Diagram.drawio** |

**Method Specification Forms**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Method Name:** | **Class Name:** | | | **ID:** | |
| **Contract ID:** | **Programmer:** | | | **Date Due:** | |
| **Programming Language: Language name** | | | | | |
| **Triggers/Events:** | | | | | |
| **Arguments Received:**  **Data Type:** | | **Notes:** | | | |
|  | |  | | | |
|  | |  | | | |
|  | |  | | | |
|  | |  | | | |
|  | |  | | | |
| **Messages Sent & Arguments Passed:**  **ClassName.MethodName:** | | | **Data Type:** | | **Notes:** |
|  | | |  | |  |
|  | | |  | |  |
|  | |  |
|  | |  |
|  | | |  | |  |
|  | | |  | |  |
|  | |  |
|  | |  |
| **Arguments Returned:**  **Data Type:** | | **Notes:** | | | |
|  | |  | | | |
| **Algorithm Specification:** | | | | | |
| **Misc. Notes:** | | | | | |

**Method Contracts**

|  |  |  |
| --- | --- | --- |
| **Method Name:** | **Class Name:** | **ID:** |
| **Clients (Consumers):** | | |
| **Associated Use Cases:** | | |
| **Description of Responsibilities:** | | |
| **Arguments Received:** | | |
| **Type of Value Returned:** | | |
| **Preconditions:** | | |
| **Postconditions:** | | |

**Chapter Eight**

**Data Management Layer Design**

**Object-Persistence Formats**

**Mapping PD Objects to Single Inheritance–Based OODBMS Schema**

**Mapping PD Objects to ORDBMS Schema**

**Mapping PD Objects to RDBMS Schema**

**Optimized ERs (Normalization or Denormalization)**

**Data Storage Size**

**Nonfunctional Requirements Design**

**Object-Persistence Formats**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data** | **Type** | **Use** | **Suggested Format** | **Reason**  **(from your project view)** |
|  |  |  |  |  |
|  |  |  |  |  |

**Mapping PD Objects to Single Inheritance–Based OODBMS Schema**

|  |
| --- |
|  |

**Mapping PD Objects to ORDBMS Schema**

|  |
| --- |
|  |

**Mapping PD Objects to RDBMS Schema**

|  |
| --- |
|  |

**Optimized ERs (Normalization or Denormalization)**

|  |
| --- |
| NNF |
|  |

|  |
| --- |
| NNF |
|  |

**Data Storage Size**

**(in case of normalization or denormalization)**

|  |  |
| --- | --- |
| **Field** | **Average Size** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| **Record Size** |  |
| **Overhead** |  |
| **Total Record Size** |  |
| **Initial Table Size** |  |
| **Initial Table Volume** |  |
| **Growth Rate/Month** |  |
| **Table Volume @ 3 years** |  |

**Nonfunctional Requirements Design**

**Chapter Nine**

**Human Computer Interaction Layer Design**

**Use Scenarios**

**WND**

**Interface Design Prototyping**

**Nonfunctional Requirements Design**

**Use Scenarios**

|  |
| --- |
| **Use scenario:** |
|  |

**WND**

|  |
| --- |
|  |

**Interface Design Prototyping**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **□** | **Storyboard** | **□** | **Windows Layout Diagram** | **□** | **HTML Prototype** | **□** | **Language Prototype** |
|  | | | | | | | |

**Nonfunctional Requirements Design**

**Project name:**

**date:**

**Chapter Ten**

**Physical Architecture Layer Design**

**Deployment Diagram**

**Nonfunctional Requirements Selection**

**Deployment Diagram of Network Model**

**Deployment Diagram of Layers**

**Hardware and Software Specification**

**Deployment Diagram**

|  |
| --- |
|  |

**Nonfunctional Requirements Selection**

|  |  |  |
| --- | --- | --- |
| **Type of Requirement** | **Definition** | **Nonfunctional Requirement** |
| **Operational Requirements** | | |
|  |  |  |
|  |  |  |
|  |  |  |
| **Performance Requirements** | | |
|  |  |  |
|  |  |  |
|  |  |  |
| **Security Requirements** | | |
|  |  |  |
|  |  |  |
|  |  |  |
| **Cultural/Political Requirements** | | |
|  |  |  |
|  |  |  |
|  |  |  |

**Deployment Diagram of Network Model**

|  |
| --- |
|  |

**Deployment Diagram of Layers**

|  |
| --- |
|  |

**Hardware and Software Specification**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Standard Client** | **Standard Web Server** | **Standard Application Server** | **Standard Database Server** |
| **Operating System** |  |  |  |  |
| **Special Software** |  |  |  |  |
| **Hardware** |  |  |  |  |
| **Network** |  |  |  |  |

**Chapter Eleven**

**Construction**

**Test Plan**

**Class Test Plan**

**Class Invariant Test Specification**

**Documentation Topic**

**Test Plan**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Stage** | **Web Interface** | **System Management** | **System Interfaces** |
|  |  |  |  |
|  |  |  |  |

**Class Test Plan**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Class Test Plan** | | | | **Page \_\_\_\_\_\_ of \_\_\_\_\_\_** | |
| **Class Name:** | | | **Version Number:** | **CRC Card ID:** | |
| **Tester:** | | | **Date Designed:** | **Date Conducted:** | |
| **Class Objective:** | | | | | |
| **Associated Contract IDs:** |  |  | | |  |
| **Associated Use Case IDs:** |  |  | | |  |
| **Associated Superclass(es):** |  |  | | |  |
| **Testing Objectives:** | | | | | |
| **Walkthrough Test Requirements:** | | | | | |
| **Invariant-Based Test Requirements:** | | | | | |
| **State-Based Test Requirements:** | | | | | |
| **Contract-Based Test Requirements:** | | | | | |

**Class Invariant Test Specification**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Class Invariant Test Specification** | | | | | | **Page \_\_\_\_\_\_ of \_\_\_\_\_\_** | |
| **Class Name:** | | | **Version Number:** | | | **CRC Card ID:** | |
| **Tester:** | | | **Date Designed:** | | | **Date Conducted:** | |
| **Testing Objectives:** | | | | | | | |
| **Test Cases:** | | | | | | | |
| **Invariant**  **Description** | **Original**  **Attribute Value** | **Event** | | **New Attribute Value** | **Expected Result** | | **Result P/F** |
| **Attribute Name:** | | | | | | | |
|  |  |  | |  |  | |  |
|  |  |  | |  |  | |  |
|  |  |  | |  |  | |  |
| **Attribute Name:** | | | | | | | |
|  |  |  | |  |  | |  |
|  |  |  | |  |  | |  |
|  |  |  | |  |  | |  |
| **Attribute Name:** | | | | | | | |
|  |  |  | |  |  | |  |
|  |  |  | |  |  | |  |
|  |  |  | |  |  | |  |

**Documentation Topic**

**Reference documents**

**procedures manuals**

**Documentation navigation controls**