**University Management System for Automating Academic Activities**



A Project submitted to the Department of Computer Science and Engineering,

Hajee Mohammad Danesh Science and Technology University

Course Title: Software Engineering

Course Code: CSE 305

|  |  |
| --- | --- |
| **Submitted To,**  Pankaj Bhowmik  Lecturer  Department of Computer Science and Engineering | **Submitted By,**  Md Shohag Showdagor  Student ID: 2102041  Level 3, Semester I |

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

HAJEE MOHAMMAD DANESH SCIENCE AND TECHNOLOGY UNIVERSITY, DINAJPUR-5200, BANGLADESH

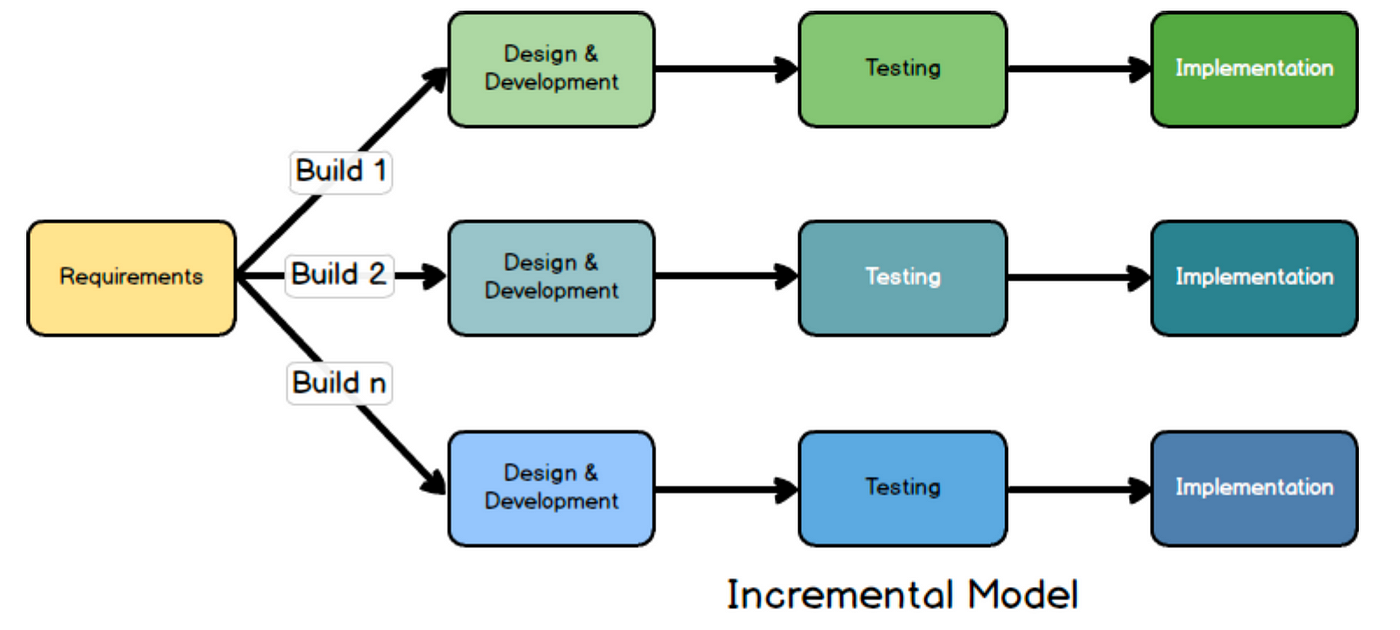
**University Management System for Automating Academic Activities**

Objective

The primary objective of this project is to design and develop a **University Management System (UMS)** that automates core academic activities, such as:

* Managing assignments and submissions.
* Generating automatic class schedules.
* Allowing deans to assign teachers to courses.
* Facilitating grading and publishing results.

The system will enhance administrative efficiency, improve student and faculty experiences, and streamline academic workflows. The **Incremental Model** is employed as the software development methodology, ensuring a systematic and phased delivery of features while maintaining the flexibility to incorporate stakeholder feedback.



**Why the Incremental Model?**

The Incremental Model is well-suited for this project due to its ability to:

* **Deliver High-Priority Features Quickly**: Each module is developed and delivered independently, enabling early adoption of critical functionalities such as assignment management or timetable generation.
* **Incorporate Feedback Effectively**: Feedback from stakeholders (students, faculty, administrators) after each increment ensures continuous improvement.
* **Reduce Risk**: Dividing development into smaller increments allows for early detection and resolution of issues.
* **Ensure Scalability**: Future modules can be added without disrupting existing functionalities

**Project Scope**

The **University Management System** will cater to the following primary users:

1. **Students**: Access assignments, view results, and check class schedules.
2. **Teachers**: Grade assignments, upload results, and manage their class schedules.
3. **Deans/Administrators**: Assign courses to teachers, manage timetables, and oversee academic processes.

The system will also include future scalability options, such as integrating attendance tracking, analytics, and notification systems.

**Implementation Using the Incremental Model**

**Phase 1: Requirement Analysis**

* **Stakeholder Input**:
  + Conduct interviews with students, teachers, and administrative staff to gather detailed requirements.
  + Identify pain points in current academic workflows and prioritize features based on their impact.
* **Requirement Specification**:
  + Create a detailed Software Requirement Specification (SRS) document outlining:
    - Functional requirements (e.g., assignment submission, timetable generation).
    - Non-functional requirements (e.g., security, performance, scalability).
  + Categorize features into increments for phased delivery.

Phase 2: System Design

* **Architectural Blueprint**:
  + Develop a modular architecture ensuring flexibility and scalability.
  + Design the database to handle user data, schedules, assignments, and results efficiently.
* **Module Planning**:
  + Plan the sequence of increments, starting with high-priority functionalities.
  + Define the dependencies between modules to streamline integration.

**Phase 3: Incremental Development**

Each increment focuses on developing and delivering a specific set of functionalities:

1. **Increment 1: Assignment Management System**
   * Features:
     + Students can submit assignments online.
     + Teachers can view, grade, and provide feedback on submissions.
   * Deliverables:
     + User interfaces for students and teachers.
     + Back-end processing for storing and managing submissions.
   * Early Benefits:
     + Immediate reduction in administrative workload related to paper-based assignments.
2. **Increment 2: Automatic Class Routine Generator**
   * Features:
     + Automated scheduling of classes based on:
       - Room availability.
       - Teacher preferences.
       - Student batch sizes.
     + Administrative dashboard for managing schedules.
   * Deliverables:
     + Algorithm for conflict-free schedule generation.
     + Interactive calendar for students and faculty.
   * Early Benefits:
     + Reduces manual errors in timetable creation.
3. **Increment 3: Teacher-Course Assignment System**
   * Features:
     + Deans can assign teachers to courses based on workload and preferences.
     + Overview of teacher workload distribution for transparency.
   * Deliverables:
     + A dashboard for course assignment and management.
   * Early Benefits:
     + Streamlined allocation of teaching responsibilities.
4. **Increment 4: Result Management System**
   * Features:
     + Teachers can upload grades, which are automatically calculated into final results.
     + Students can view results through a secure portal.
     + Option for re-evaluation requests.
   * Deliverables:
     + Grading portal for teachers.
     + Secure student access to results.
   * Early Benefits:
     + Accelerated and transparent result publication.

**Future Increments (Planned):**

* Attendance tracking for classes.
* Analytics dashboards for academic performance monitoring.
* Notification and communication systems for announcements.

**Testing and Validation**

* **Unit Testing**:
  + Test individual modules for functionality and correctness.
* **Integration Testing**:
  + Ensure seamless interaction between modules like timetable generation and course assignments.
* **User Testing**:
  + Involve stakeholders after each increment to gather feedback and validate usability.
* **Performance Testing**:
  + Optimize for high user loads, particularly during result publication or assignment deadlines.

**Deployment and Maintenance**

* **Incremental Deployment**:
  + Deploy each increment upon completion, allowing users to benefit from early functionalities.
* **Feedback Mechanism**:
  + Incorporate user feedback from live usage into subsequent increments.
* **System Updates**:
  + Provide regular updates to fix bugs and enhance features.

**Key Features**

1. **Assignment Management**:
   * Simplifies the submission and grading process.
   * Reduces paperwork and improves communication.
2. **Automatic Class Scheduling**:
   * Ensures efficient timetable creation with minimal manual effort.
3. **Teacher-Course Assignment**:
   * Facilitates equitable and transparent workload distribution.
4. **Result Publication**:
   * Accelerates grading and provides secure access to results.

**Benefits of the Incremental Model**

1. **User-Centric Design**:
   * Continuous feedback ensures the system aligns with user needs.
2. **Early Value Delivery**:
   * High-priority features become available early in the development process.
3. **Risk Management**:
   * Smaller, iterative development cycles reduce the chances of failure.
4. **Scalability**:
   * The modular approach ensures future features can be seamlessly integrated.

By adopting the Incremental Model, this project emphasizes delivering a robust, scalable, and user-friendly system that automates academic activities while remaining adaptable to the evolving needs of universities.