***Institut Supérieur d'Informatique et Mathématiques de Monastir***

***Département d'Informatique***

***Section : CPI2 AU : 2023-2024***

***Mini-projet programmation Orientée Objet JAVA***

**Problème: Gestion des projets de Fin d'Etudes (PFE).**

**Description générale du problème :**

Chaque étudiant à la fin de sa formation doit réaliser un PFE. Un PFE peut être affecté à un seul ou deux étudiant(s).

***Les groupes :*** il existe trois types:

* Licence: informatique, électronique...
* Mastère : Recherche, Professionnel.
* Ingénieur: Informatique et électronique.

***Les enseignants:*** un enseignant est un encadreur pédagogique. Un enseignant peut encadrer un ou plusieurs étudiants / PFE.

***Jurys:*** est formé par un seul président et les trois membres qui sont: rapporteur, examinateur et encadreur(s) (on peut avoir des invités qui sont des encadreurs de la société).

***Planification des soutenances:*** chaque soutenance doit être planifiée à une date , heure et un locale . Une fois la soutenance est passée elle sera validée ou non et on lui affectera une note.

**Step 1: Requirements Gathering and Analysis**

1. Understand the problem statement thoroughly.
2. Identify all the functionalities and features required.
3. Define the data entities (Student, Group, Teacher, Jury, Presentation) and their attributes.
4. Determine relationships between entities (e.g., student-group relationship, teacher-student relationship).
5. Decide on the user interface requirements (CLI, GUI).

**Step 2: Design**

1. Design the class hierarchy:
   * Identify classes and their relationships.
   * Determine inheritance, composition, and aggregation relationships.
2. Design class attributes and methods:
   * Define attributes for each class.
   * Determine the behavior (methods) for each class.
3. Design database schema (if applicable):
   * Define tables and their relationships.
   * Determine primary and foreign keys.

**Step 3: Implementation**

1. Set up your development environment:
   * Install Java Development Kit (JDK) if not already installed.
   * Set up a project structure.
   * Configure any dependencies (e.g., JDBC for database connectivity).
2. Implement the class hierarchy:
   * Start with the basic classes (e.g., Student, Teacher).
   * Implement methods to manipulate data within each class.
3. Implement business logic:
   * Implement functionalities like assigning students to groups, scheduling presentations, etc.
4. Implement user interface:
   * If CLI, create text-based menus and prompts for user interaction.
   * If GUI, design and implement the graphical interface using Java Swing or JavaFX.
5. Implement database integration (if applicable):
   * Set up a database (e.g., MySQL, PostgreSQL).
   * Write code to connect to the database and perform CRUD operations.
6. Implement error handling and validation:
   * Validate user input to ensure data integrity.
   * Handle exceptions gracefully.

**Step 4: Testing**

1. Write unit tests for each class and method to ensure correctness.
2. Perform integration testing to verify interactions between different modules.
3. Test edge cases and boundary conditions.
4. Debug and fix any issues that arise during testing.

**Step 5: Documentation**

1. Document your code:
   * Provide comments for classes, methods, and complex algorithms.
   * Use Javadoc for generating API documentation.
2. Write a user manual:
   * Explain how to install and run the program.
   * Provide instructions for using each feature.
3. Write a project report:
   * Summarize the problem statement and your solution.
   * Discuss challenges faced and lessons learned.
   * Include screenshots (if applicable) to illustrate your program in action.

**Step 6: Review and Refinement**

1. Review your code for clarity, efficiency, and adherence to best practices.
2. Refactor code to improve readability and maintainability.
3. Get feedback from peers or mentors and incorporate any suggestions for improvement.

**Step 7: Deployment**

1. Package your application for distribution (if applicable).
2. Provide instructions for installation and usage.
3. Deploy your application to a suitable environment (local machine, server).

**Step 8: Maintenance and Updates**

1. Monitor for any issues or bugs reported by users.
2. Release updates and patches as necessary.
3. Continuously improve your application based on user feedback and changing requirements