Created by: Matthew Veroutis & Matthew Macri

Project Proposal: School Registration System

Objective

To develop a School Registration System that allows students, teachers, and administrators to interact with course management. The system will enable students to register for courses, view their schedules, and manage their academic information, while teachers and administrators will have their own distinct functionalities. The project will focus on fundamental data structures, unit testing, and object-oriented principles.

Overview

The system will provide the following functionalities:

- Students: Register for courses, view schedules, and drop courses.
- Teachers: View course details and class rosters.
- Administrators: Add, update, and remove courses, and manage student enrollments.

The project will use data structures such as `ArrayList`, `LinkedList`, `Stack`, and `Queue`, along with interfaces like `Comparable` and `Comparator`. Unit tests will ensure the correctness of operations, and Maven will handle dependency management.

Key Features

1. Student Portal:
    - Register and drop courses (add/drop functionality).
    - View current course schedule.
2. Teacher Portal:
    - View course details such as enrolled students and class schedules.
3. Administrator Portal:
    - Add, update, or remove courses.
    - Manage student enrollment.
4. Data Structures:
    - Use ArrayList and LinkedList for course management.
    - Implement Queue to manage student registration requests.
    - Apply Comparable and Comparator interfaces for sorting courses.

5. Unit Testing:
    - Ensure code quality by creating unit tests for each data structure operation (e.g., add, remove).
    - Use JUnit for testing.
6. External Libraries:
    - Use Maven for dependency management and external libraries.
7. Generic Classes:
    - Implement a generic data structure such as `LinkedList` to handle different types of data (e.g., courses and users).
8. Stream Processing:
    - Use Lambda expressions and stream processing to filter courses by time, professor, or level.

Technologies

- Java: Core programming language.
- JUnit: For unit testing and quality assurance.
- Maven: For dependency management and building the project.
- Java Streams & Lambda Expressions: To simplify code and make data operations more efficient.
- JDBC (optional): For database connectivity (SQLite/Oracle) if required for advanced features.
- Git: Version control for code collaboration and management.

Timeframe

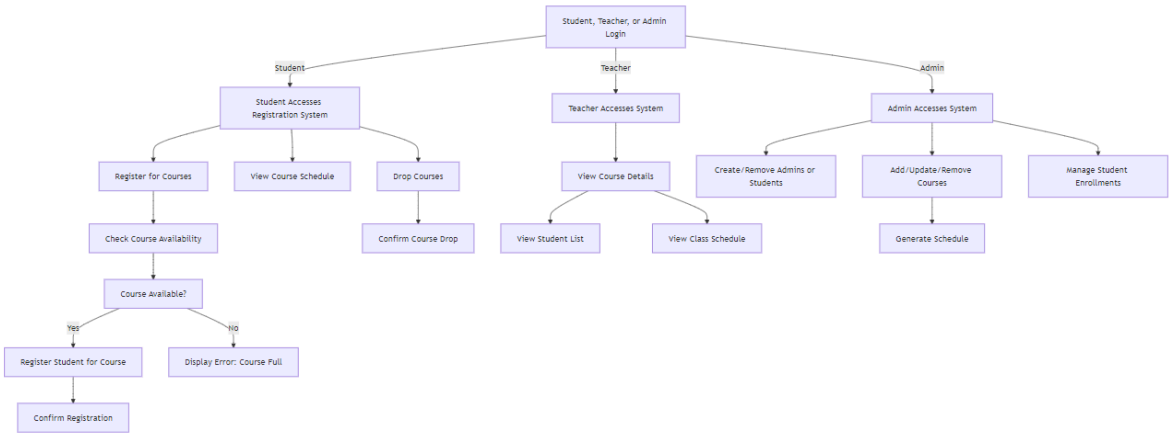
- Week 1-2: Implement core features (student registration, course management).
- Week 3-4: Implement generic data structures and sorting mechanisms (using Comparable and Comparator).
- Week 5: Add unit testing and finalize the project.

Expected Outcome

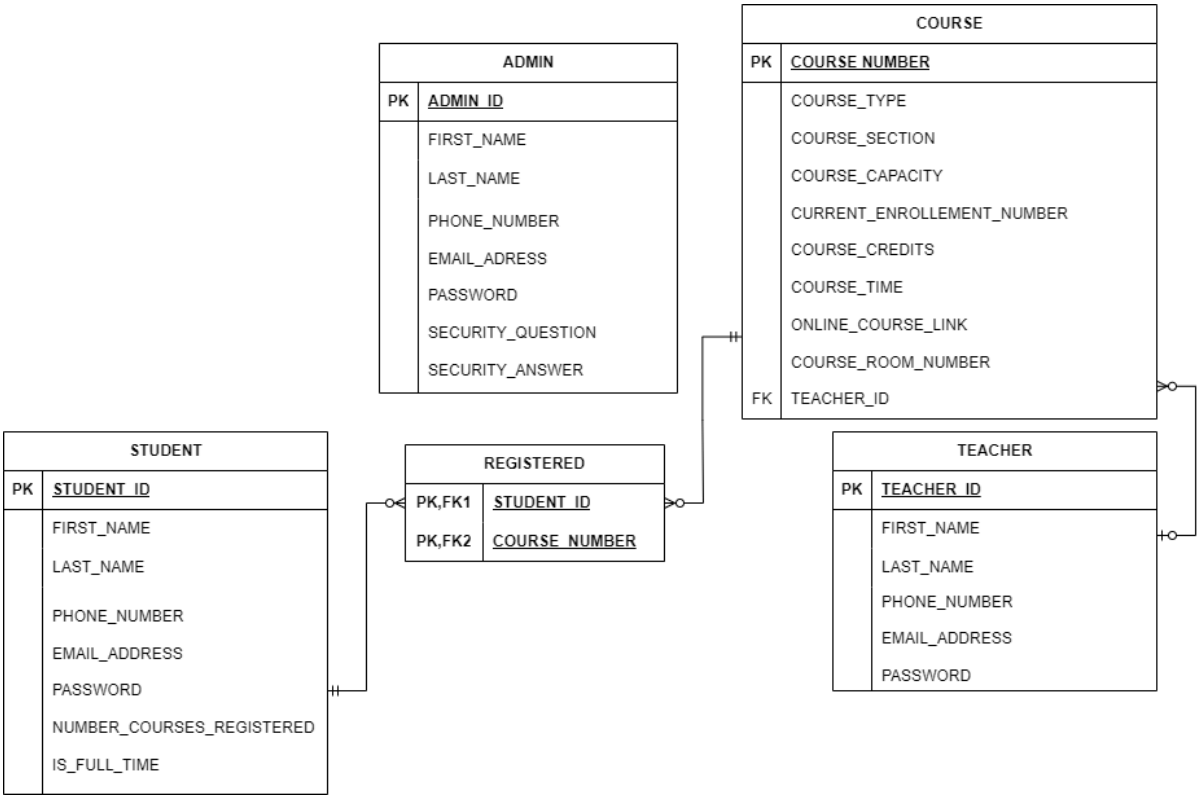
A functional School Registration System that allows students, teachers, and administrators to interact with course data. The project will demonstrate strong data structures, unit testing, and stream processing concepts, while maintaining a clean and maintainable design.

Created by: Matthew Veroutis & Matthew Macri

Activity Diagram:

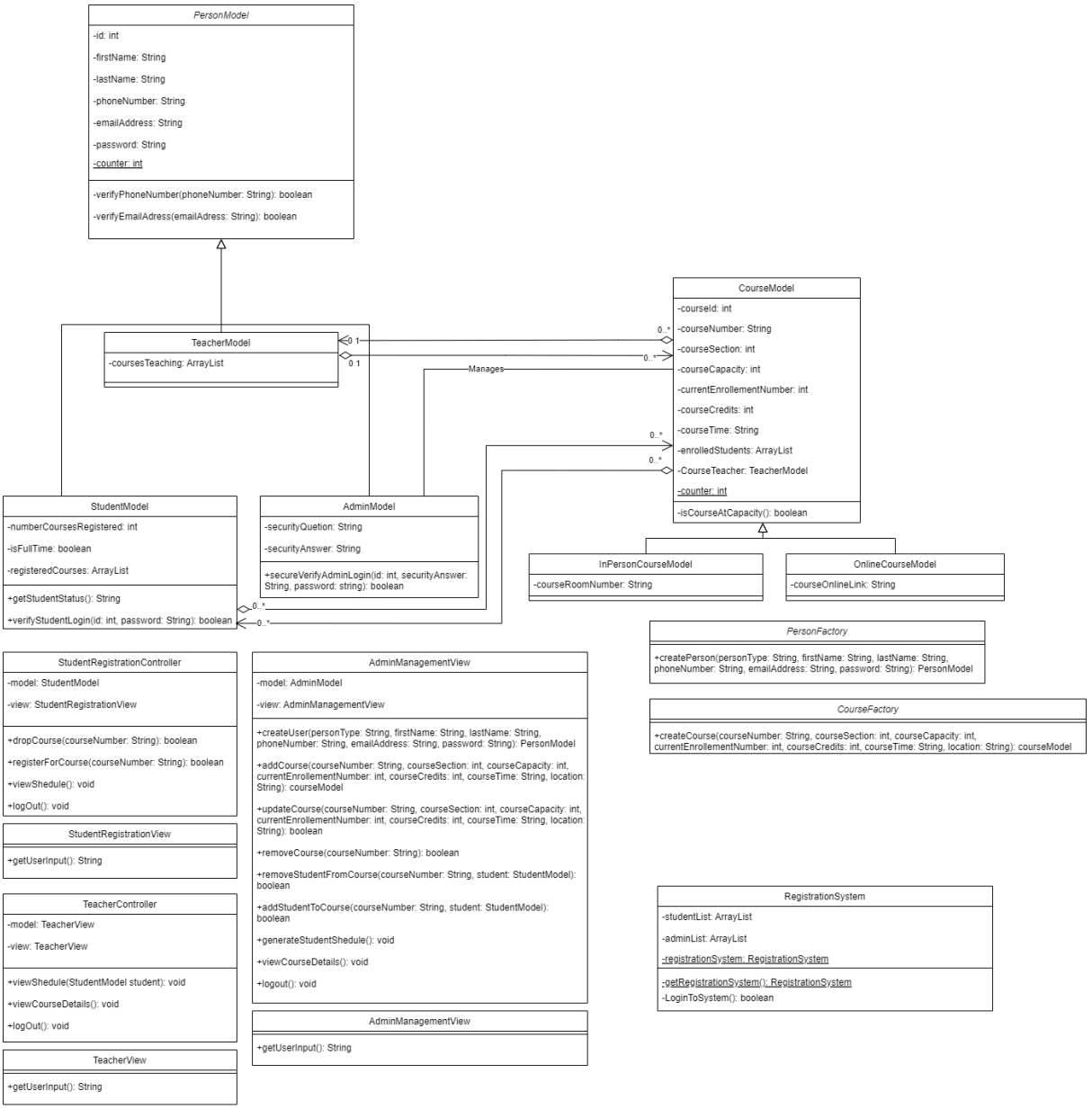


Entity Relationship Diagram:



Created by: Matthew Veroutis & Matthew Macri

Class Diagram:



GitHub Repository Link: https://github.com/MatthewMacri/Programming-Patterns-Project.git