FileHandler

+STUDENT_FILE: String = "students.txt" {readOnly}
+SUBJECT_FILE: String = "subjects.txt" {readOnly}
+COURSE_FILE: String = "courses.txt" {readOnly}
+REG_FILE: String = "registrations.txt" {readOnly}

+loadStudents(String fileName): List<Student>
+loadSubjects(String fileName): List<Subject>
+loadCourseSections(String fileName):
List<CourseSection>
+loadRegistrations(String fileName):
List<Registration>
+saveDataToFile(List<? extends FileSerializable> list,
String fileName): void

CourseRegistrationSystem

-studentManager: StudentManager ;
-subjectManager: SubjectManager;
-courseManager: CourseManager;
-registrationManager: RegistrationManager;
-fileHandler: FileHandler;

+main(String[] args): void
+run(): void
-handleSaveData(): void
-handleStudentManagement(): void
-handleRegisterForStudent(): void
-handleSubjectManager(): void
-handleRegistrationManager(): void

Validator

-sc: Scanner

+getString(String inputMsg, String
errorMsg): String
+ getString(String prompt, String errorMsg,
String regex): String
+getAnInteger(String inputMsg, String
errorMsg, int lowerBound, int upperBound):
int
+getYesNo()

Menu

+showMenu(): void
+showStudentManagementMenu(): void
+showSubjectManagementMenu(): void
+showCourseManagementMenu(): void

<<interface class>>
Identifiable

+getId(): String

<<interface class>>
FileSerializable

+toFileString(): String

<>
Person

#fullName: String
#email: String

Student

-studentId: String
-major: String
-status: StudentStatus

+Student()
+Student(studentId: String, fullName: String,
major: String,email: String, status:
StudentStatus)
+getters/setters
+toString(): String
+toFileString(): String
+getId(): String

Subject

-subjectId: String
-subjectName: String
-credit: int
-prerequisiteSubjectIds: List<String>

+Subject()
+Subject(subjectId: String, subjectName:
String, credit: int)
+getters/setters
+addPrerequisite(String subjectId): void
+toString(): String
+toFileString(): String
+getId(): String

CourseSection

-courseSectionId: String
-subjectId: String
-semester: int
-maxStudents: int
-currentStudentCount: int
-dayOfWeek: DayOfWeek
-startSlot: int
-endSlot: int

+CourseSection()
+CourseSection(String courseSectionId:
String, subjectId: String, semester: int,
maxStudents: int, int currentStudentCount:
int, dayOfWeek: DayOfWeek, startSlot: int,
endSlot: int)
+getters/setters

+isFull(): boolean
+incrementStudentCount(): void
+decrementStudentCount(): void
+toString(): String

+toFileString(): String
+getId(): String

Registration

-studentId: String
-courseSectionId: String
-grade: double
-status: RegistrationStatus
+RegistrationStatus: {ENROLLED, PASSED,
FAILED, WITHDRAWN}

+Registration()
+Registration(studentId: String,
courseSectionId: String, grade: double,
RegistrationStatus status)
+getters/setters
+updateStatusByGrade(): void
+toString(): String
+toFileString(): String
+getId(): String

<<<enumeration>>
RegistrationStatus

ENROLLED,
PASSED,
FAILED,
WITHDRAWN

<<<enumeration>>
StudentStatus

ACTIVE,
INACTIVE,
GRADUATED

<<<enumeration>>
DayOfWeek

MON,
TUE,
WED,
THU,
FRI,
SAT

<>
Management

#list: List<T>

+Management (List<T> initialList)
+getAll(): List<T>
+findById(String id): T
+add(T item): boolean
+delete(T item): boolean
+update(T itemToUpdate): boolean

<<interface class>>
Displayable

+displayAll(): void

StudentManager

+findByName(name: String): List<Student>
+sortByName(): List<Student>
-getLastName(fullName: String): String
+displayAll(): void

SubjectManager

+ getSectionsBySubject(subjectId: String):
List<CourseSection>
+ displayAll() : void

CourseManager

+ displayAll() : void

RegistrationManager

+ registerCourse(studentId : String, courseSectionId : String) : boolean
- calculateCurrentCredits(studentId : String, semester : int) : boolean
+ withdrawCourse(studentId : String, courseSectionId : String) : boolean
+ getRegistrationsByStudent(studentId : String) : List<Registration>

+ getRegistrationsByCourse(sectionId : String) : List<Registration>
+ getStudentsByCourseSection(String courseSectionId): List<Student>
+ getStudentsBySubject(subjectId: String): List<Student>
+ getSubjectsByStudent(studentId: String): List<Subject>

+ getStudentsSortedByOverallGPA(): List<Student>
+ getStudentsSortedBySubjectGPA(String subjectId): List<Student>
- findGradeForSubject(studentId: String, targetSubjectId: String): double
+ calculateOverallGPA(studentId: String): double

+ calculateSemesterGPA(studentId: String, semester: int): double
+ displayAll() : void