After Class Diagram

Util

Menu

- OutputMenu(): void
- listStudents(Map<String, Student>): void
- listCourses(Map<String, Course>): void
- menuLoop(Map<String, Student>, Map<String, Course>): void
- createMenu(Map<String, Student>, Map<String, Course>): void

Output

+ print(String): void
+ println(String): void
# Start(): void
# Exit(): void

extends

UI

~ audit: Audit

+ addStudentUI(Scanner, Map<String, Student>): void
+ addCourseUI(Scanner, Map<String, Course>): void
+ enrollUI(Scanner, Map<String, Course>): void
+ dropUI(Scanner, Map<String, Course>): void
+ listStudents(Map<String, Student>): void
+ listCourses(Map<String, Course>): void

Parser

- split(String): String[]
+ parseStudent(String): Student
+ parseCourse(String): Course
+ parseEnrollment(String): String[]

EnrollmentException

- audit: Audit {readOnly}

+ EnrollmentException(String):

ValidationException

- audit: Audit {readOnly}

+ ValidationException(String):

FileIntegrity

- checkAndCreateCSV(String): void

Validate

- result: Boolean
- CHAR_LIMIT: int {readOnly}
+ audit: Audit

+ Validate_String(String, String): void
+ getResult(): Boolean

ValidateCourse

- CODE_MATCH: String {readOnly}
- TITLE_MATCH: String {readOnly}
- CAPACITY_MATCH: String {readOnly}

+ Validate_Code(String): boolean
+ Validate_Title(String): boolean
+ Validate_Capacity(String): boolean

ValidateStudent

- BANNERID_MATCH: String {readOnly}
- NAME_MATCH: String {readOnly}
- EMAIL_MATCH: String {readOnly}

+ ValidateID(String): boolean
+ ValidateName(String): boolean
+ ValidateEmail(String): boolean

Model

Course

- code: String
- title: String
- capacity: int
- roster: List<String>
- waitlist: List<String>

+ getCode(): String
+ getTitle(): String
+ getCapacity(): int
+ getRoster(): List<String>
+ getWaitlist(): List<String>
+ setCode(String): void
+ setTitle(String): void
+ setCapacity(int): void
+ setRoster(List<String>): void
+ setWaitlist(List<String>): void

Student

- id: String
- name: String
- email: String

+ toString(): String
+ getId(): String
+ getName(): String
+ getEmail(): String
+ setId(String): void
+ setName(String): void
+ setEmail(String): void

App

Main

- STUDENTS_CSV: String {readOnly}
- COURSES_CSV: String {readOnly}
- ENROLLMENT_CSV: String {readOnly}
- InCSV: InCSV {readOnly}
- OUTCSV: InCSV {readOnly}
- checkFileIntegrity: FileIntegrity {readOnly}

+ main(String[]): void
- seedDemoData(Map<String, Student>, Map<String, Course>): void

Repo

<<Interface>> CourseRepository

~ loadCourses(Map<String, Course>, String): void
~ saveCourses(Map<String, Course>, String): void

<<Interface>> EnrollmentRepository

~ loadEnrollment(Map<String, Course>, String): void
~ saveEnrollment(Map<String, Course>, String): void

<<Interface>> StudentRepository

~ loadStudents(Map<String, Student>, String): void
~ saveStudents(Map<String, Student>, String): void

Service

RegisrationService

- audit: Audit {readOnly}
- parser: Parser {readOnly}

+ loadAll(Map<String, Student>, Map<String, Course>, InCSV): void
+ saveAll(Map<String, Student>, Map<String, Course>, OutCSV): void
+ loadStudents(Map<String, Student>, String): void
+ saveStudents(Map<String, Student>, String): void
+ loadCourses(Map<String, Course>, String): void
+ saveCourse(Map<String, Course>, String): void
+ loadEnrollments(Map<String, Course>, String): void
+ saveEnrollments(Map<String, Course>, String): void

Audit

~ auditLog: List<String>

+ add(String): void
+ write_logfile(): void