|  |
| --- |
| **Student** |
| -studentID: String  -firstName: String  -lastName: String  -middleName: Char  -studentMonthOfAdmittance: String  -studentYearOfAdmittance: int  -studentMajor: Major  -studentTranscript: LinkedList<Course>  -schedule: Schedule  -planner: Planner |
| +Student(studentID: String)  -editSchedule()  -editPlanner()  -updateTranscript() |

|  |
| --- |
| **Course** |
| -department: String  -courseID: String  -courseName: String  -hasLab: Boolean  -units: Integer  -prerequisites: ArrayList<String[]>  -description: String  /estimatedDifficulty: Double  -hoursSurveyData: LinkedHashSet<Integer, Integer>  -gradeDistribution: LinkedHashSet<String, Integer> |
| +Course(department:String, courseID:String, courseName:String, hasLab:Boolean, units:Integer)  -prereqCheck(ArrayList<String[]>): Boolean  -inputSurveyData(ArrayList<Integer>)  -inputGradeData(ArrayList<String>) |

|  |
| --- |
| **Section** |
| -sectionID: String  -sectionName: String  -component: String //e.g. LEC, LAB, ACT  -gradingOption: String //e.g. GRD - Graded, CNC - Credit/No Credit  -startDate: String  -endDate: String  -daysOfWeek: String  -startTime: String  -endTime: String  -location: String  -notes: ArrayList<String[]> //e.g. 5207 - Registration limited to majors requiring the course, E - Applied Arts and Sciences  -instructor: String //default is Staff |
| +Section(sectionID: String, sectionName: String, component: String, gradingOption: String) |

|  |
| --- |
| **Major** |
| -majorID: String  -majorName: String  -majorReq: Graph<Courses[]> |
| +Major(majorID: String, majorName: String)  -addMajorReq(courseID: String)  +displayMajorReq() |

|  |
| --- |
| **Planner** |
| -PlannerUnits: PlannerUnit[]  -goalGraduationSemeter: String  -goalGraduationYear: Integer //default is four years after the student was admitted to the college |
| +Planner()  +editPlanner(semester: String, year: Integer)  -orderChronologically() //ensures PlannerUnits are stored chronologically to cut down on time to access the earliest available semester |

|  |
| --- |
| **PlannerUnit** |
| -semester: String  -year: Integer  -maxUnits: Integer  -difficultyRating: Double  -courses: Course[] |
| +PlannerUnit(semester: String, year: Integer)  +addCourse(courseID: String): Boolean  +removeCourse(courseID: String): Boolean  +swapCourses(addCourseID: String,  removeCourseID: String): Boolean |

|  |
| --- |
| **Schedule** |
| -semester: String  -year: Integer  -targetDifficulty: Double  -totalDifficulty: Double  -maxUnits: Integer  -totalUnits: Integer  -numSections: Integer  -sections: Section[] |
| +Schedule(semester: String, year: Integer)  +addSection(sectionID: String)  +deleteSection(sectionID: String) |

|  |
| --- |
| **UI** |
| +Planner: ArrayList<String[]>  +Schedule: ArrayList<String[]> //these arrays serve as output only |
| +UI(studentID: String)  +viewPlanner()  +viewSchedule()  -assignColorCode() |

|  |
| --- |
| **AI** |
| None |
| +generatePlanner(Student: student): Planner  +generateSchedule(Planner: Planner): Schedule  +suggestAlternative(): LinkedList<Section[]> //if a user wishes to modify the generated schedule, this function suggests list of alternative courses that will fulfill the same requirement; e.g. if it’s for a GE fulfillment  +calculateDifficultyRating(Planner: Planner)  +checkChangeOfMajor(studentTranscript: LinkedList<Course>, majorID: String): Integer //returns a percentage that rates similarities between the courses a student has taken to the courses required to fulfill a major (e.g. 50 would mean a student has taken 50% of the courses required for the specified major)  +optimizeChangeInMajor(studentTranscript: LinkedList<Course>) //compares the courses a student has taken to all majors offered by the college and ranks majors from highest similarities (that is, a high integer returned from the checkChangeOfMajor function) to lowest  +impactedGradWarning(studentID: String, Planner: Planner)  -createChain(courseID: String)  -addAll(courseID: String) //takes one course and links ALL pre- and post-requisites  -findPrereq(String: courseName)  -compareMajor(majorID: String) //this would compare the chain with the major to know where to cut off; e.g. certain majors only require you to go up to COMP182 so it would remove COMP282, COMP380, etc. from the chain  -compareTranscript(studentTranscript: LinkedList<Course>) //same as above, except it cuts off links in the chain if the student already took the class  -linkExists(courseID: String) //checks to see if a specific course is in the chain; e.g. a COMP 110 to COMP 490 chain wouldn’t include a HIST 101 link  -cutLinks(courseID: String, prerequisite: Boolean) //this is the function that would sever links. It begins at the relevant course and either removes the entire left side of the chain (Boolean = true), or the entire right side (Boolean = false); e.g. cutLinks(COMP182, false) would cut ALL the postrequisites for 182 while leaving 182 intact  -countLinks(): Integer //more links means a longer chain  -displayChain() |