DataConstants
...

DataWriter
+ getUsers(): ArrayList<Users>
+ getCourses():
ArrayList<Course>

+ saveAllStudents()
+ saveAllAdvisors()

DataLoader
+ saveUsers(): void;
+ saveCourses(): void;

+ getAllStudents()
+ getAllAdvisors()

UI
...

...

Application
- user: User
- userList: UserList
- courseList: CourseList

+ Application()
+ login(String username, String password): User
+ signUp(String username, String password): User
+ addCourse(ArrayList<String courseComments, int courseNumber, CourseIdentifiers courseID, String instructor, ArrayList<String> prereq, String description, int totalSeats, int creditWorth, Code courseCode): boolean
+ addStudentCourse(ArrayList<String courseComments, int courseNumber, CourseIdentifiers courseID, String instructor, ArrayList<String> prereq, String description, int totalSeats, int creditWorth, Code courseCode, double Grade, boolean completed, boolean enrolled)
+ addUser(String firstName, String lastName, int ID, String email, String username, String password)
+ viewMajorMap(major Major)
+ addStudent(double gpa, boolean minor, ArrayList<String> StudentComments, ArrayList<Course> enrolledCourses, ArrayList<Achievement> achievements, advisor Advisor, boolean riskFailing, double hoursCompleted, String studentID)
+ addFaculty(String officeHours, ArrayList<Student> studentList)
+ addAdvisor(String office, String officeHours, String phoneNumber, ArrayList<Student> adviseeList, String schoolOfFocus, ArrayList<Apointment> appointments)
+ findCourse(String keyword): ArrayList<Course>
+ bookAppointment(String time, String location, Advisor advisor): void
+ viewWhatIf(int projectedGraduationYear, int additionalHoursNeeded, Major newMajor): ArrayList<Course>
+ sendStudentComment(Student currentStudent, String comment): String
+ addAchievement(Achievement achievement): void
+ addEnrolledCourse(EnrolledCourse enrolledCourse): void
+ viewAdvisees(ArrayList<Student> advisees) : void
+ viewAdvisee(): Student
+ viewStudent(Student currentStudent): String
+ getAppointments(): ArrayList<Appointment>
+ addAdvisee(Student advisee): void
+ addAppointment(Advisor advisor, String time, String location): boolean
+ viewStudents(ArrayList<Student> students) : void
+ viewCourse(Couse course): String

MajorList
- _ majorList: MajorList
- majors: ArrayList<Major>

-MajorList()
+ getInstance(): MajorList majorList

UserList
- userList: UserList
- users: ArrayList<User>

-UserList()
+ getInstance(): UserList userList
+ getUser(String userName): User

CourseList
-courseList: CourseList
- courses: ArrayList<course>

-CourseList()
+getCourse(String CourseName) ArrayList<courses>

Major
- majorID: UUID
- title: String
- requiredCourses: ArrayList<Course>
- completedCourses: ArrayList<Course>
- hoursRequired: int
- hoursCompleted: int
- progression: double

+ Major(String title, ArrayList<Course> courses, ArrayList<Course> completedCourses, int hoursRequired, int hoursCompleted, int progression)
+ addRequiredCourse(Course course): void
+ addCompletedCourse(Course course): void
+ calculateProgression(): void
+

User
- firstName: String
- lastName: String
- userID: UUID
- email: String
- username: String
- password: String
- type: int

+ User(String firstName, String lastName, String ID, String email, String username, String password, int type)
+ viewStudentProfile(Student student): String

Course
courseComments: ArrayList<String>
- courseNumber: int
- courseID: UUID
- courseCode: String
- courseName: String
- instructor: Faculty
- prerequisites: ArrayList<Course>
- description: String
- totalSeats: int
- creditWorth: int
- corequesites: ArrayList<Course>
- applicationArea: boolean
- carolinaCore: boolean
- elective: boolean
- grade: Double
- completed: Boolean
- enrolled: Boolean

+ Course(String courseName, ArrayList<String courseComments, int courseNumber, CourseIdentifiers courseID, String instructor, ArrayList<String> prereq, String description, int totalSeats, int creditWorth, Code courseCode, double Grade, boolean completed, boolean enrolled)
+ Course(String courseName, ArrayList<String> courseComments, int courseNumber, CourseIdentifiers courseID, String instructor, ArrayList<String> prereq, String description, int totalSeats, int creditWorth, Code courseCode)
+ addPrequisite(Course prereq): void
+ addCorequesites(Course coreq): void

Student
- major: Major
- minor: boolean
studentComments: ArrayList<String>
- achievements: ArrayList<Achievement>
- gpa: double
- enrolledCourses: ArrayList<Course>
- advisor: Advisor
- riskFailing: boolean
- hoursCompleted: double
- studentID: String

+ Student(double gpa, boolean minor, ArrayList<String> StudentComments, ArrayList<Course> enrolledCourses, ArrayList<Achievement> achievements, advisor Advisor, boolean riskFailing, double hoursCompleted, String studentID).
+ sendAdvisorComment(Advisor advisor): void
+ sendInstructorComment(Advisor advisor): void
+ viewStudentProfile(Student student): String
+addEnrolledCourse(EnrolledCourse enrolledCourse): void
+ whatIfScenario(int projectedGraduationYear, int addiitonalHoursNeeded, Major newMajor)
+ makeAppointment(String time, String location, Advisor advisor): void
+ calculateGPA(ArrayList<CompletedCourses> completedCourses): double

Faculty
- officeHours: String
- studentList: ArrayList<Student>

+ Faculty(String officeHours, ArrayList<Student> studentList)
+ writeCourseComment(Course course)
+ viewStudents(ArrayList<Student> students) : void
+ viewStudentProfile(Student student): String
+ addStudent(Studnet student): void

Advisor
- office: String
- officeHours: String
- phoneNumber: String
- adviseeList: ArrayList<Student>
- schoolOfFocus: String
appointments: ArrayList<Appointment>

+ Advisor(String office, String officeHours, String phoneNumber, ArrayList<Student> adviseeList, String schoolOfFocus, ArrayList<Apointment> appointments)
+ writeStudentComment(Student student): String
+ viewAdvisees(ArrayList<Student> advisees) : void
+ viewStudent(Student student): String
+ getAppointments(): ArrayList<Appointment>
+ addAdvisee(Student advisee): void
+ addApointment(Appointment appointment): void

StudentSearch
- searchField: String

+ StudentSearch(Sting searchField)
+ SearchByName(String searchField): ArrayList<Student>
+ SearchByID String searchField): ArrayList<Student>
+ SearchByUsername String searchField): ArrayList<Student>

Appointment
- advisorID: UUID
- time: String
- location: String

+ Appointment(Advisor advisor, String time, String location)

Achievement
- achievementID: UUID
- name: String
- type: String
- benefits: String
- semester: String
- description String
- scholarship: Boolean

+ Achievement(String name, String type, String benefits, String semester, String description)
+ hasScholarship(scholarship): Boolean