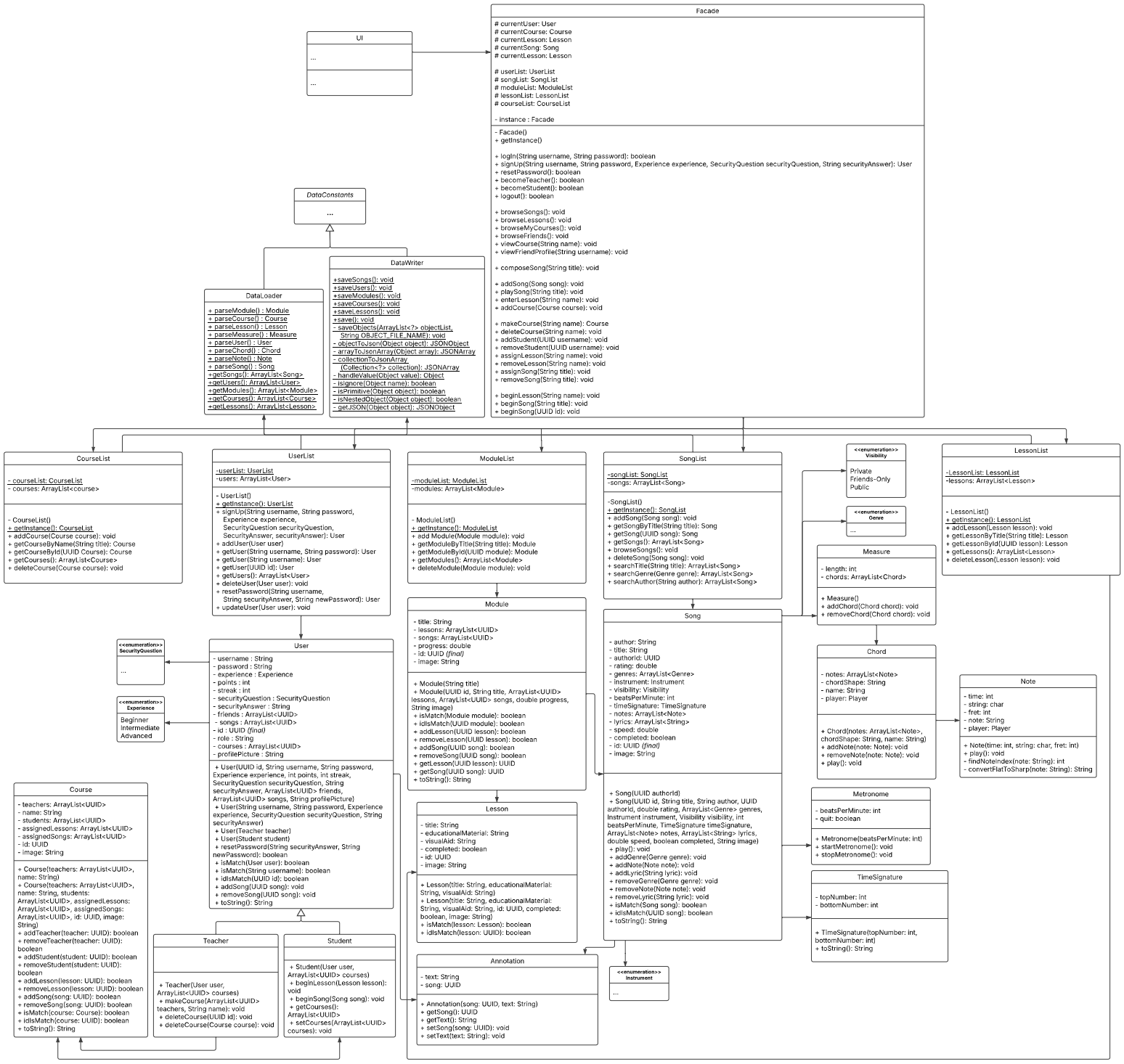
UI
...
...
Facade
currentUser: User
currentCourse: Course
currentLesson: Lesson
currentSong: Song
currentLesson: Lesson
userList: UserList
songList: SongList
moduleList: ModuleList
lessonList: LessonList
courseList: CourseList
- instance : Facade
- Facade()
+ getInstance()
+ logIn(String username, String password): boolean
+ signUp(String username, String password, Experience experience, SecurityQuestion securityQuestion, String securityAnswer): User
+ resetPassword(): boolean
+ becomeTeacher(): boolean
+ becomeStudent(): boolean
+ logout(): boolean
+ browseSongs(): void
+ browseLessons(): void
+ browseMyCourses(): void
+ browseFriends(): void
+ viewCourse(String name): void
+ viewFriendProfile(String username): void
+ composeSong(String title): void
+ addSong(Song song): void
+ playSong(String title): void
+ enterLesson(String name): void
+ addCourse(Course course): void
+ makeCourse(String name): Course
+ deleteCourse(String name): void
+ addStudent(UUID username): void
+ removeStudent(UUID username): void
+ assignLesson(String name): void
+ removeLesson(String name): void
+ assignSong(String title): void
+ removeSong(String title): void
+ beginLesson(String name): void
+ beginSong(String title): void
+ beginSong(UUID id): void
DataConstants
...
DataLoader
+ parseModule() : Module
+ parseCourse() : Course
+ parseLesson() : Lesson
+ parseMeasure() : Measure
+ parseUser() : User
+ parseChord() : Chord
+ parseNote() : Note
+ parseSong() : Song
+getSongs(): ArrayList<Song>
+getUsers(): ArrayList<User>
+getModules(): ArrayList<Module>
+getCourses(): ArrayList<Course>
+getLessons(): ArrayList<Lesson>
DataWriter
+saveSongs(): void
+saveUsers(): void
+saveModules(): void
+saveCourses(): void
+saveLessons(): void
+save(): void
- saveObjects(ArrayList<?> objectList, String OBJECT_FILE_NAME): void
- objectToJson(Object object): JSONObject
- arrayToJsonArray(Object array): JSONArray
- collectionToJsonArray (Collection<?> collection): JSONArray
- handleValue(Object value): Object
- isIgnore(Object name): boolean
- isPrimitive(Object object): boolean
- isNestedObject(Object object): boolean
- getJSON(Object object): JSONObject
CourseList
- courseList: CourseList
- courses: ArrayList<course>
- CourseList()
+ getInstance(): CourseList
+ addCourse(Course course): void
+ getCourseByName(String title): Course
+ getCourseById(UUID Course): Course
+ getCourses(): ArrayList<Course>
+ deleteCourse(Course course): void
UserList
-userList: UserList
-users: ArrayList<User>
- UserList()
+ getInstance(): UserList
+ signUp(String username, String password, Experience experience, SecurityQuestion securityQuestion, SecurityAnswer, securityAnswer): User
+ addUser(User user)
+ getUser(String username, String password): User
+ getUser(String username): User
+ getUser(UUID id): User
+ getUsers(): ArrayList<User>
+ deleteUser(User user): void
+ resetPassword(String username, String securityAnswer, String newPassword): User
+ updateUser(User user): void
ModuleList
-moduleList: ModuleList
-modules: ArrayList<Module>
- ModuleList()
+ getInstance(): ModuleList
+ add Module(Module module): void
+ getModuleByTitle(String title): Module
+ getModuleById(UUID module): Module
+ getModules(): ArrayList<Module>
+ deleteModule(Module module): void
SongList
-songList: SongList
-songs: ArrayList<Song>
-SongList()
+ getInstance(): SongList
+ addSong(Song song): void
+ getSongByTitle(String title): Song
+ getSong(UUID song): Song
+ getSongs(): ArrayList<Song>
+ browseSongs(): void
+ deleteSong(Song song): void
+ searchTitle(String title): ArrayList<Song>
+ searchGenre(Genre genre): ArrayList<Song>
+ searchAuthor(String author): ArrayList<Song>
<<enumeration>>
Visibility
Private
Friends-Only
Public
<<enumeration>>
Genre
...
LessonList
-LessonList: LessonList
-lessons: ArrayList<Lesson>
- LessonList()
+ getInstance(): LessonList
+ addLesson(Lesson lesson): void
+ getLessonByTitle(String title): Lesson
+ getLessonById(UUID lesson): Lesson
+ getLessons(): ArrayList<Lesson>
+ deleteLesson(Lesson lesson): void
Measure
- length: int
- chords: ArrayList<Chord>
+ Measure()
+ addChord(Chord chord): void
+ removeChord(Chord chord): void
Chord
- notes: ArrayList<Note>
- chordShape: String
- name: String
- player: Player
+ Chord(notes: ArrayList<Note>, chordShape: String, name: String)
+ addNote(note: Note): void
+ removeNote(note: Note): void
+ play(): void
Note
- time: int
- string: char
- fret: int
- note: String
- player: Player
+ Note(time: int, string: char, fret: int)
+ play(): void
- findNoteIndex(note: String): int
- convertFlatToSharp(note: String): String
Module
- title: String
- lessons: ArrayList<UUID>
- songs: ArrayList<UUID>
- progress: double
- id: UUID (final)
- image: String
+ Module(String title)
+ Module(UUID id, String title, ArrayList<UUID> lessons, ArrayList<UUID> songs, double progress, String image)
+ isMatch(Module module): boolean
+ idIsMatch(UUID module): boolean
+ addLesson(UUID lesson): boolean
+ removeLesson(UUID lesson): boolean
+ addSong(UUID song): boolean
+ removeSong(UUID song): boolean
+ getLesson(UUID lesson): UUID
+ getSong(UUID song): UUID
+ toString(): String
Lesson
- title: String
- educationalMaterial: String
- visualAid: String
- completed: boolean
- id: UUID
- image: String
+ Lesson(title: String, educationalMaterial: String, visualAid: String)
+ Lesson(title: String, educationalMaterial: String, visualAid: String, id: UUID, completed: boolean, image: String)
+ isMatch(lesson: Lesson): boolean
+ idIsMatch(lesson: UUID): boolean
Annotation
- text: String
- song: UUID
+ Annotation(song: UUID, text: String)
+ getSong(): UUID
+ getText(): String
+ setSong(song: UUID): void
+ setText(text: String): void
Song
- author: String
- title: String
- authorId: UUID
- rating: double
- genres: ArrayList<Genre>
- instrument: Instrument
- visibility: Visibility
- beatsPerMinute: int
- timeSignature: TimeSignature
- notes: ArrayList<Note>
- lyrics: ArrayList<String>
- speed: double
- completed: boolean
- id: UUID (final)
- image: String
+ Song(UUID authorId)
+ Song(UUID id, String title, String author, UUID authorId, double rating, ArrayList<Genre> genres, Instrument instrument, Visibility visibility, int beatsPerMinute, TimeSignature timeSignature, ArrayList<Note> notes, ArrayList<String> lyrics, double speed, boolean completed, String image)
+ play(): void
+ addGenre(Genre genre): void
+ addNote(Note note): void
+ addLyric(String lyric): void
+ removeGenre(Genre genre): void
+ removeNote(Note note): void
+ removeLyric(String lyric): void
+ isMatch(Song song): boolean
+ idIsMatch(UUID song): boolean
+ toString(): String
<<enumeration>>
Instrument
...
Metronome
- beatsPerMinute: int
- quit: boolean
+ Metronome(beatsPerMinute: int)
+ startMetronome(): void
+ stopMetronome(): void
TimeSignature
- topNumber: int
- bottomNumber: int
+ TimeSignature(topNumber: int, bottomNumber: int)
+ toString(): String
<<enumeration>>
SecurityQuestion
...
<<enumeration>>
Experience
Beginner
Intermediate
Advanced
User
- username : String
- password : String
- experience : Experience
- points : int
- streak : int
- securityQuestion : SecurityQuestion
- securityAnswer : String
- friends : ArrayList<UUID>
- songs : ArrayList<UUID>
- id : UUID (final)
- role : String
- courses : ArrayList<UUID>
- profilePicture : String
+ User(UUID id, String username, String password, Experience experience, int points, int streak, SecurityQuestion securityQuestion, String securityAnswer, ArrayList<UUID> friends, ArrayList<UUID> songs, String profilePicture)
+ User(String username, String password, Experience experience, SecurityQuestion securityQuestion, String securityAnswer)
+ User(Teacher teacher)
+ User(Student student)
+ resetPassword(String securityAnswer, String newPassword): boolean
+ isMatch(User user): boolean
+ isMatch(String username): boolean
+ idIsMatch(UUID id): boolean
+ addSong(UUID song): void
+ removeSong(UUID song): void
+ toString(): String
Course
- teachers: ArrayList<UUID>
- name: String
- students: ArrayList<UUID>
- assignedLessons: ArrayList<UUID>
- assignedSongs: ArrayList<UUID>
- id: UUID
- image: String
+ Course(teachers: ArrayList<UUID>, name: String)
+ Course(teachers: ArrayList<UUID>, name: String, students: ArrayList<UUID>, assignedLessons: ArrayList<UUID>, assignedSongs: ArrayList<UUID>, id: UUID, image: String)
+ addTeacher(teacher: UUID): boolean
+ removeTeacher(teacher: UUID): boolean
+ addStudent(student: UUID): boolean
+ removeStudent(student: UUID): boolean
+ addLesson(lesson: UUID): boolean
+ removeLesson(lesson: UUID): boolean
+ addSong(song: UUID): boolean
+ removeSong(song: UUID): boolean
+ isMatch(course: Course): boolean
+ idIsMatch(course: UUID): boolean
+ toString(): String
Teacher
+ Teacher(User user, ArrayList<UUID> courses)
+ makeCourse(ArrayList<UUID> teachers, String name): void
+ deleteCourse(UUID id): void
+ deleteCourse(Course course): void
Student
+ Student(User user, ArrayList<UUID> courses)
+ beginLesson(Lesson lesson): void
+ beginSong(Song song): void
+ getCourses(): ArrayList<UUID>
+ setCourses(ArrayList<UUID> courses): void