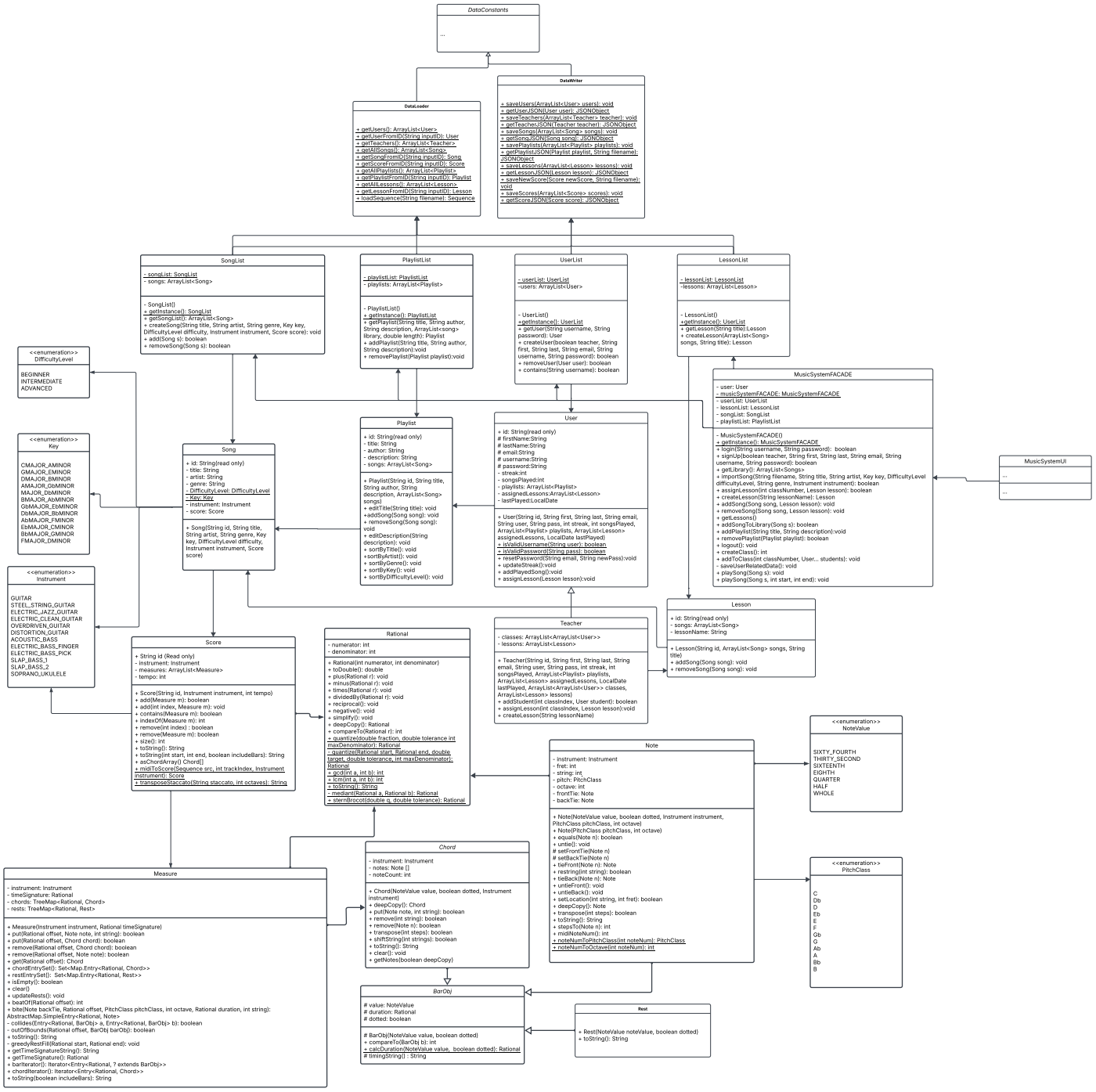
DataConstants
...
DataLoader
+ getUsers(): ArrayList<User>
+ getUserFromID(String inputID): User
+ getTeachers(): ArrayList<Teacher>
+ getAllSongs(): ArrayList<Song>
+ getSongFromID(String inputID): Song
+ getScoreFromID(String inputID): Score
+ getAllPlaylists(): ArrayList<Playlist>
+ getPlaylistFromID(String inputID): Playlist
+ getAllLessons(): ArrayList<Lesson>
+ getLessonFromID(String inputID): Lesson
+ loadSequence(String filename): Sequence
DataWriter
+ saveUsers(ArrayList<User> users): void
+ getUserJSON(User user): JSONObject
+ saveTeachers(ArrayList<Teacher> teacher): void
+ getTeacherJSON(Teacher teacher): JSONObject
+ saveSongs(ArrayList<Song> songs): void
+ getSongJSON(Song song): JSONObject
+ savePlaylists(ArrayList<Playlist> playlists): void
+ getPlaylistJSON(Playlist playlist, String filename): JSONObject
+ saveLessons(ArrayList<Lesson> lessons): void
+ getLessonJSON(Lesson lesson): JSONObject
+ saveNewScore(Score newScore, String filename): void
+ saveScores(ArrayList<Score> scores): void
+ getScoreJSON(Score score): JSONObject
SongList
- songList: SongList
- songs: ArrayList<Song>
- SongList()
+ getInstance(): SongList
+ getSongList(): ArrayList<Song>
+ createSong(String title, String artist, String genre, Key key, DifficultyLevel difficulty, Instrument instrument, Score score): void
+ add(Song s): boolean
+ removeSong(Song s): boolean
PlaylistList
- playlistList: PlaylistList
- playlists: ArrayList<Playlist>
- PlaylistList()
+ getInstance(): PlaylistList
+ getPlaylist(String title, String author, String description, ArrayList<song> library, double length): Playlist
+ addPlaylist(String title, String author, String description):void
+ removePlaylist(Playlist playlist):void
UserList
- userList: UserList
-users: ArrayList<User>
- UserList()
+getInstance(): UserList
+ getUser(String username, String password): User
+ createUser(boolean teacher, String first, String last, String email, String username, String password): boolean
+ removeUser(User user): boolean
+ contains(String username): boolean
LessonList
- lessonList: LessonList
-lessons: ArrayList<Lesson>
- LessonList()
+getInstance(): UserList
+ getLesson(String title):Lesson
+ createLesson(ArrayList<Song> songs, String title): Lesson
<<enumeration>>
DifficultyLevel
BEGINNER
INTERMEDIATE
ADVANCED
MusicSystemFACADE
- user: User
- musicSystemFACADE: MusicSystemFACADE
- userList: UserList
- lessonList: LessonList
- songList: SongList
- playlistList: PlaylistList
- MusicSystemFACADE()
+ getInstance(): MusicSystemFACADE
+ login(String username, String password): boolean
+ signUp(boolean teacher, String first, String last, String email, String username, String password): boolean
+ getLibrary(): ArrayList<Songs>
+ importSong(String filename, String title, String artist, Key key, DifficultyLevel difficultyLevel, String genre, Instrument instrument): boolean
+ assignLesson(int classNumber, Lesson lesson): boolean
+ createLesson(String lessonName): Lesson
+ addSong(Song song, Lesson lesson): void
+ removeSong(Song song, Lesson lesson): void
+ getLessons()
+ addSongToLibrary(Song s): boolean
+ addPlaylist(String title, String description):void
+ removePlaylist(Playlist playlist): boolean
+ logout(): void
+ createClass(): int
+ addToClass(int classNumber, User... students): void
- saveUserRelatedData(): void
+ playSong(Song s): void
+ playSong(Song s, int start, int end): void
MusicSystemUI
...
...
<<enumeration>>
Key
CMAJOR_AMINOR
GMAJOR_EMINOR
DMAJOR_BMINOR
AMAJOR_GbMINOR
MAJOR_DbMINOR
BMAJOR_AbMINOR
GbMAJOR_EbMINOR
DbMAJOR_BbMINOR
AbMAJOR_FMINOR
EbMAJOR_CMINOR
BbMAJOR_GMINOR
FMAJOR_DMINOR
Song
+ id: String(read only)
- title: String
- artist: String
- genre: String
- DifficultyLevel: DiffIcultyLevel
- Key: Key
- instrument: Instrument
- score: Score
+ Song(String id, String title, String artist, String genre, Key key, DiffIcultyLevel difficulty, Instrument instrument, Score score)
Playlist
+ id: String(read only)
- title: String
- author: String
- description: String
- songs: ArrayList<Song>
+ Playlist(String id, String title, String author, String description, ArrayList<Song> songs)
+ editTitle(String title): void
+addSong(Song song): void
+ removeSong(Song song): void
+ editDescription(String description): void
+ sortByTitle(): void
+sortByArtist(): void
+ sortByGenre(): void
+ sortByKey(): void
+ sortByDifficultyLevel(): void
User
+ id: String(read only)
# firstName:String
# lastName:String
# email:String
# username:String
# password:String
- streak:int
- songsPlayed:int
- playlists: ArrayList<Playlist>
- assignedLessons:ArrayList<Lesson>
- lastPlayedLocalDate
+ User(String id, String first, String last, String email, String user, String pass, int streak, int songsPlayed, ArrayList<Playlist> playlists, ArrayList<Lesson> assignedLessons, LocalDate lastPlayed)
+ isValidUsername(String user): boolean
+ isValidPassword(String pass): boolean
+ resetPassword(String email, String newPass):void
+ updateStreak():void
+ addPlayedSong():void
+ assignLesson(Lesson lesson):void
<<enumeration>>
Instrument
GUITAR
STEEL_STRING_GUITAR
ELECTRIC_JAZZ_GUITAR
ELECTRIC_CLEAN_GUITAR
OVERDRIVEN_GUITAR
DISTORTION_GUITAR
ACOUSTIC_BASS
ELECTRIC_BASS_FINGER
ELECTRIC_BASS_PICK
SLAP_BASS_1
SLAP_BASS_2
SOPRANO_UKULELE
Score
+ String id (Read only)
- instrument: Instrument
- measures: ArrayList<Measure>
- tempo: int
+ Score(String id, Instrument instrument, int tempo)
+ addMeasure m): boolean
+ add(int index, Measure m): void
+ contains(Measure m): boolean
+ indexOf(Measure m): int
+ remove(int index) : boolean
+ remove(Measure m): boolean
+ size(): int
+ toString(): String
+ toString(int start, int end, boolean includeBars): String
+ asChordArray() Chord[]
+ midiToScore(Sequence src, int trackIndex, Instrument instrument): Score
+ transposeStaccato(String staccato, int octaves): String
Rational
- numerator: int
- denominator: int
+ Rational(int numerator, int denominator)
+ toDouble(): double
+ plus(Rational r): void
+ minus(Rational r): void
+ times(Rational r): void
+ dividedBy(Rational r): void
+ reciprocal(): void
+ negative(): void
+ simplify(): void
+ deepCopy(): Rational
+ compareTo(Rational r): int
+ quantize(double fraction, double tolerance int maxDenominator): Rational
- quantize(Rational start, Rational end, double target, double tolerance, int maxDenominator): Rational
+ gcd(int a, int b): int
+ lcm(int a, int b): int
+ toString(): String
- mediant(Rational a, Rational b): Rational
+ sternBrocot(double q, double tolerance): Rational
Teacher
- classes: ArrayList<ArrayList<User>>
- lessons: ArrayList<Lesson>
+ Teacher(String id, String first, String last, String email, String user, String pass, int streak, int songsPlayed, ArrayList<Playlist> playlists, ArrayList<Lesson> assignedLessons, LocalDate lastPlayed, ArrayList<ArrayList<User>> classes, ArrayList<Lesson> lessons)
+ addStudent(int classIndex, User student): boolean
+ assignLesson(int classindex, Lesson lesson):void
+ createLesson(String lessonName)
Lesson
+ id: String(read only)
- songs: ArrayList<Song>
- lessonName: String
+ Lesson(String id, ArrayList<Song> songs, String title)
+ addSong(Song song): void
+ removeSong(Song song): void
<<enumeration>>
NoteValue
SIXTY_FOURTH
THIRTY_SECOND
SIXTEENTH
EIGHTH
QUARTER
HALF
WHOLE
Note
- instrument: Instrument
- fret: int
- string: int
- pitch: PitchClass
- octave: int
- frontTie: Note
- backTie: Note
+ Note(NoteValue value, boolean dotted, Instrument instrument, PitchClass pitchClass, int octave)
+ Note(PitchClass pitchClass, int octave)
+ equals(Note n): boolean
+ untie(): void
# setFrontTie(Note n)
# setBackTie(Note n)
+ tieFront(Note n): Note
+ restring(int string): boolean
+ tieBack(Note n): Note
+ untieFront(): void
+ untieBack(): void
+ setLocation(int string, int fret): boolean
+ deepCopy(): Note
+ transpose(int steps): boolean
+ toString(): String
+ stepsTo(Note n): int
+ midiNoteNum(): int
+ noteNameToPitchClass(int noteNum): PitchClass
+ noteNumToOctave(int noteNum): int
<<enumeration>>
PitchClass
C
Db
D
Eb
E
F
Gb
G
Ab
A
Bb
B
Measure
- instrument: Instrument
- timeSignature: Rational
- chords: TreeMap<Rational, Chord>
- rests: TreeMap<Rational, Rest>
+ Measure(Instrument instrument, Rational timeSignature)
+ put(Rational offset, Note note, int string): boolean
+ put(Rational offset, Chord chord): boolean
+ remove(Rational offset, Chord chord): boolean
+ remove(Rational offset, Note note): boolean
+ get(Rational offset): Chord
+ chordEntrySet(): Set<Map.Entry<Rational, Chord>>
+ restEntrySet(): Set<Map.Entry<Rational, Rest>>
+ isEmpty(): boolean
+ clear()
+ updateRests(): void
+ beatOf(Rational offset): int
+ bite(Note backTie, Rational offset, PitchClass pitchClass, int octave, Rational duration, int string): AbstractMap.SimpleEntry<Rational, Note>
- collides(Entry<Rational, BarObj> a, Entry<Rational, BarObj> b): boolean
- outOfBounds(Rational offset, BarObj barObj): boolean
+ toString(): String
- greedyRestFill(Rational start, Rational end): void
+ getTimeSignatureString(): String
+ getTimeSignature(): Rational
+ barIterator(): Iterator<Entry<Rational, ? extends BarObj>>
+ chordIterator(): Iterator<Entry<Rational, Chord>>
+ toString(boolean includeBars): String
Chord
- instrument: Instrument
- notes: Note []
- noteCount: int
+ Chord(NoteValue value, boolean dotted, Instrument instrument)
+ deepCopy(): Chord
+ put(Note note, int string): boolean
+ remove(int string): boolean
+ remove(Note n): boolean
+ transpose(int steps): boolean
+ shiftString(int strings): boolean
+ toString(): String
+ clear(): void
+ getNotes(boolean deepCopy)
BarObj
value: NoteValue
duration: Rational
dotted: boolean
BarObj(NoteValue value, boolean dotted)
+ compareTo(BarObj b): int
+ calcDuration(NoteValue value, boolean dotted): Rational
# timingString() : String
Rest
+ Rest(NoteValue noteValue, boolean dotted)
+ toString(): String