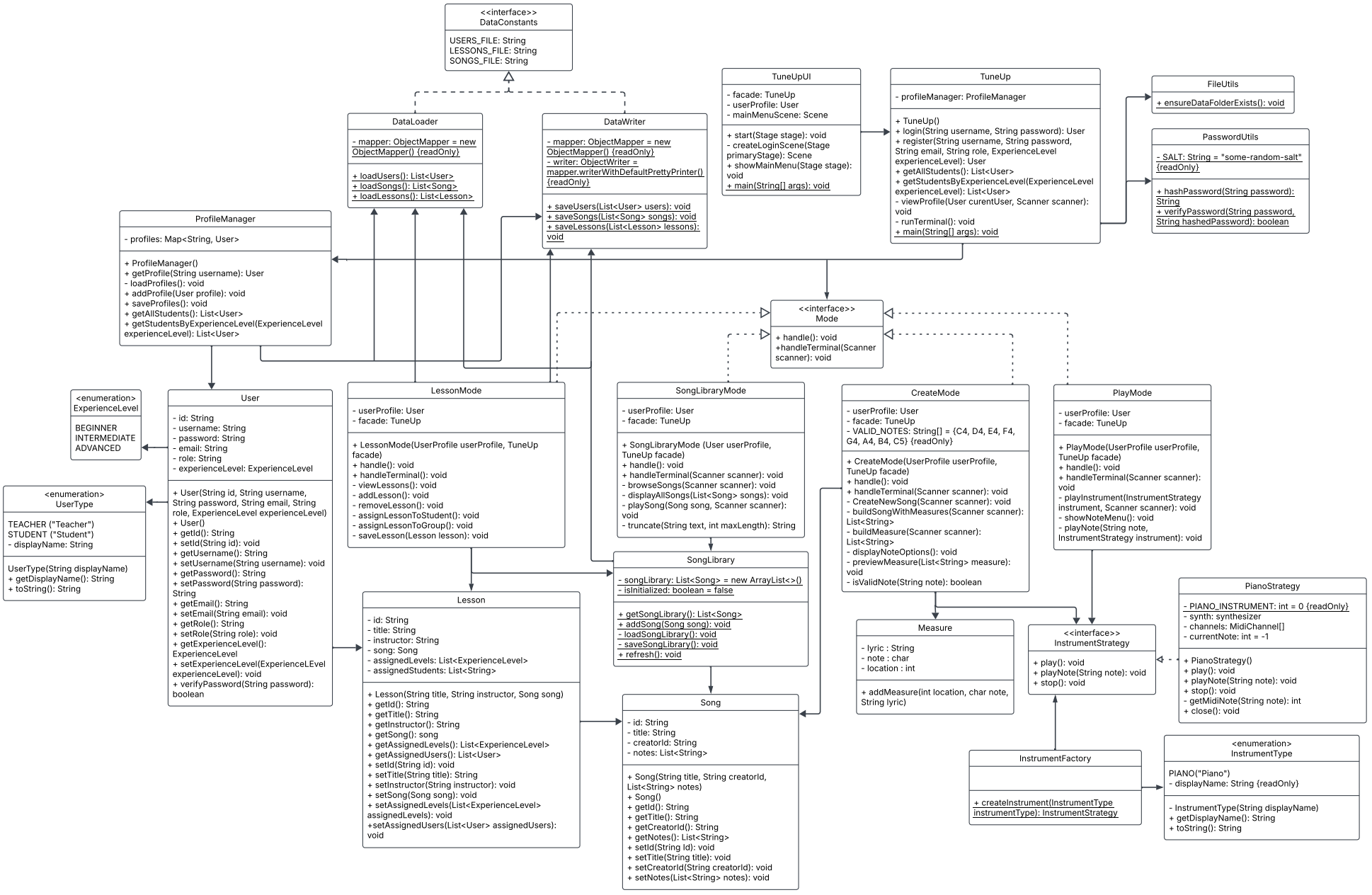
<<interface>>
DataConstants
USERS_FILE: String
LESSONS_FILE: String
SONGS_FILE: String
DataLoader
- mapper: ObjectMapper = new ObjectMapper() {readOnly}
+ loadUsers(): List<User>
+ loadSongs(): List<Song>
+ loadLessons(): List<Lesson>
DataWriter
- mapper: ObjectMapper = new ObjectMapper() {readOnly}
- writer: ObjectWriter = mapper.writerWithDefaultPrettyPrinter() {readOnly}
+ saveUsers(List<User> users): void
+ saveSongs(List<Song> songs): void
+ saveLessons(List<Lesson> lessons): void
TuneUpUI
- facade: TuneUp
- userProfile: User
- mainMenuScene: Scene
+ start(Stage stage): void
- createLoginScene(Stage primaryStage): Scene
+ showMainMenu(Stage stage): void
+ main(String[] args): void
TuneUp
- profileManager: ProfileManager
+ TuneUp()
+ login(String username, String password): User
+ register(String username, String password, String email, String role, ExperienceLevel experienceLevel): User
+ getAllStudents(): List<User>
+ getStudentsByExperienceLevel(ExperienceLevel experienceLevel): List<User>
- viewProfile(User curentUser, Scanner scanner): void
- runTerminal(): void
+ main(String[] args): void
FileUtils
+ ensureDataFolderExists(): void
PasswordUtils
- SALT: String = "some-random-salt" {readOnly}
+ hashPassword(String password): String
+ verifyPassword(String password, String hashedPassword): boolean
ProfileManager
- profiles: Map<String, User>
+ ProfileManager()
+ getProfile(String username): User
- loadProfiles(): void
+ addProfile(User profile): void
+ saveProfiles(): void
+ getAllStudents(): List<User>
+ getStudentsByExperienceLevel(ExperienceLevel experienceLevel): List<User>
<<interface>>
Mode
+ handle(): void
+handleTerminal(Scanner scanner): void
<enumeration>
ExperienceLevel
BEGINNER
INTERMEDIATE
ADVANCED
User
- id: String
- username: String
- password: String
- email: String
- role: String
- experienceLevel: ExperienceLevel
+ User(String id, String username, String password, String email, String role, ExperienceLevel experienceLevel)
+ User()
+ getId(): String
+ setId(String id): void
+ getUsername(): String
+ setUsername(String username): void
+ getPassword(): String
+ setPassword(String password): String
+ getEmail(): String
+ setEmail(String email): void
+ getRole(): String
+ setRole(String role): void
+ getExperienceLevel(): ExperienceLevel
+ setExperienceLevel(ExperienceLEvel experienceLevel): void
+ verifyPassword(String password): boolean
<enumeration>
UserType
TEACHER ("Teacher")
STUDENT ("Student")
- displayName: String
UserType(String displayName)
+ getDisplayName(): String
+ toString(): String
LessonMode
- userProfile: User
- facade: TuneUp
+ LessonMode(UserProfile userProfile, TuneUp facade)
+ handle(): void
+ handleTerminal(): void
- viewLessons(): void
- addLesson(): void
- removeLesson(): void
- assignLessonToStudent(): void
- assignLessonToGroup(): void
- saveLesson(Lesson lesson): void
SongLibraryMode
- userProfile: User
- facade: TuneUp
+ SongLibraryMode (User userProfile, TuneUp facade)
+ handle(): void
+ handleTerminal(Scanner scanner): void
- browseSongs(Scanner scanner): void
- displayAllSongs(List<Song> songs): void
- playSong(Song song, Scanner scanner): void
- truncate(String text, int maxLength): String
CreateMode
- userProfile: User
- facade: TuneUp
- VALID_NOTES: String[] = {C4, D4, E4, F4, G4, A4, B4, C5} {readOnly}
+ CreateMode(UserProfile userProfile, TuneUp facade)
+ handle(): void
+ handleTerminal(Scanner scanner): void
- CreateNewSong(Scanner scanner): void
- buildSongWithMeasures(Scanner scanner): List<String>
- buildMeasure(Scanner scanner): List<String>
- displayNoteOptions(): void
- previewMeasure(List<String> measure): void
- isValidNote(String note): boolean
PlayMode
- userProfile: User
- facade: TuneUp
+ PlayMode(UserProfile userProfile, TuneUp facade)
+ handle(): void
+ handleTerminal(Scanner scanner): void
- playInstrument(InstrumentStrategy instrument, Scanner scanner): void
- showNoteMenu(): void
- playNote(String note, InstrumentStrategy instrument): void
SongLibrary
- songLibrary: List<Song> = new ArrayList<>()
- isInitialized: boolean = false
+ getSongLibrary(): List<Song>
+ addSong(Song song): void
- loadSongLibrary(): void
- saveSongLibrary(): void
+ refresh(): void
Lesson
- id: String
- title: String
- instructor: String
- song: Song
- assignedLevels: List<ExperienceLevel>
- assignedStudents: List<String>
+ Lesson(String title, String instructor, Song song)
+ getId(): String
+ getTitle(): String
+ getInstructor(): String
+ getSong(): song
+ getAssignedLevels(): List<ExperienceLevel>
+ getAssignedUsers(): List<User>
+ setId(String id): void
+ setTitle(String title): String
+ setInstructor(String instructor): void
+ setSong(Song song): void
+ setAssignedLevels(List<ExperienceLevel> assignedLevels): void
+setAssignedUsers(List<User> assignedUsers): void
Song
- id: String
- title: String
- creatorId: String
- notes: List<String>
+ Song(String title, String creatorId, List<String> notes)
+ Song()
+ getId(): String
+ getTitle(): String
+ getCreatorId(): String
+ getNotes(): List<String>
+ setId(String Id): void
+ setTitle(String title): void
+ setCreatorId(String creatorId): void
+ setNotes(List<String> notes): void
Measure
- lyric : String
- note : char
- location : int
+ addMeasure(int location, char note, String lyric)
<<interface>>
InstrumentStrategy
+ play(): void
+ playNote(String note): void
+ stop(): void
PianoStrategy
- PIANO_INSTRUMENT: int = 0 {readOnly}
- synth: synthesizer
- channels: MidiChannel[]
- currentNote: int = -1
+ PianoStrategy()
+ play(): void
+ playNote(String note): void
+ stop(): void
- getMidiNote(String note): int
+ close(): void
InstrumentFactory
+ createInstrument(InstrumentType instrumentType): InstrumentStrategy
<enumeration>
InstrumentType
PIANO("Piano")
- displayName: String {readOnly}
- InstrumentType(String displayName)
+ getDisplayName(): String
+ toString(): String