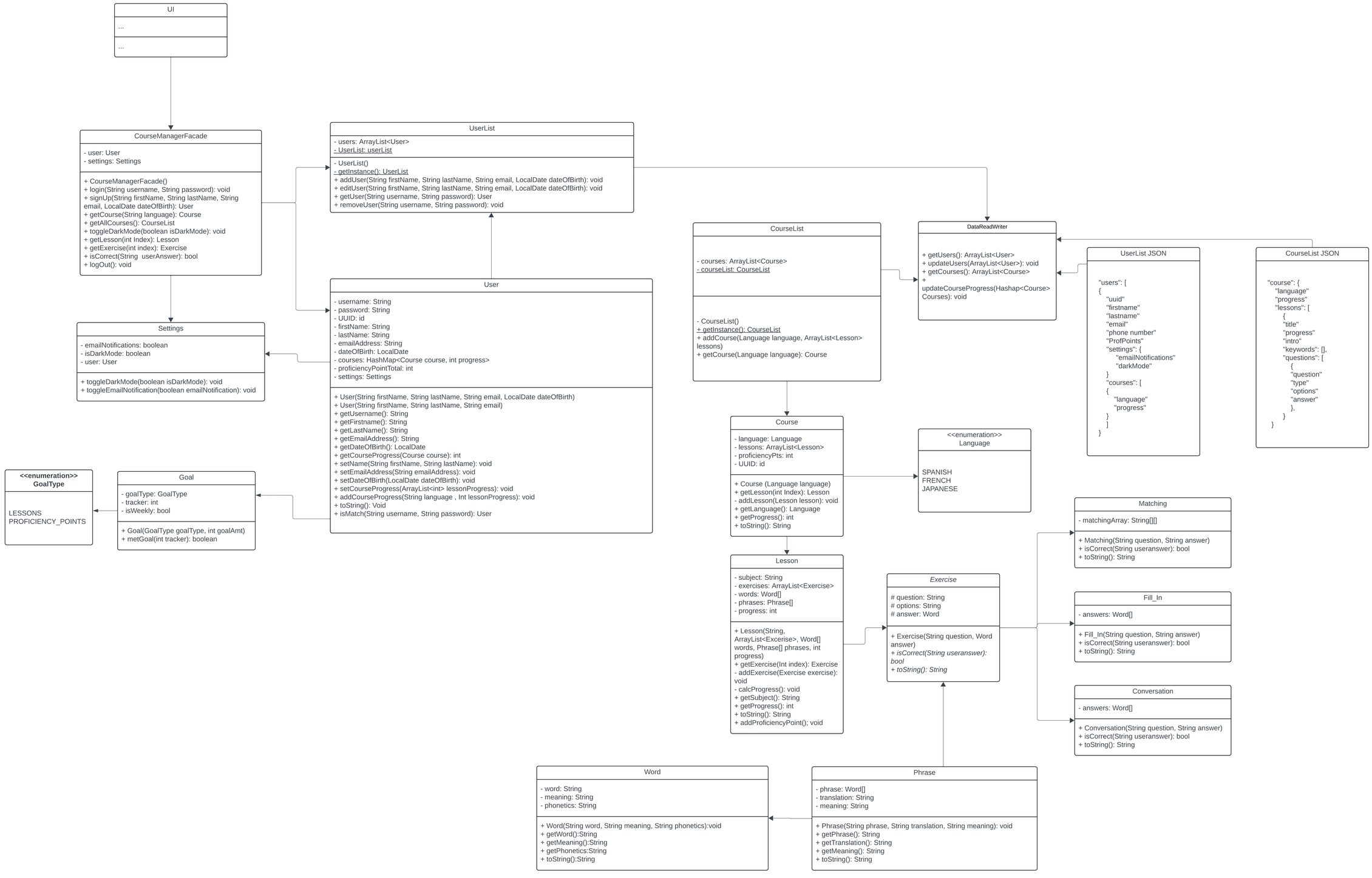
UI
...
...
CourseManagerFacade
- user: User
- settings: Settings
+ CourseManagerFacade()
+ login(String username, String password): void
+ signUp(String firstName, String lastName, String email, LocalDate dateOfBirth): User
+ getCourse(String language): Course
+ getAllCourses(): CourseList
+ toggleDarkMode(boolean isDarkMode): void
+ getLesson(int Index): Lesson
+ getExercise(int index): Exercise
+ isCorrect(String userAnswer): bool
+ logOut(): void
UserList
- users: ArrayList<User>
- UserList: userList
- UserList()
- getInstance(): UserList
+ addUser(String firstName, String lastName, String email, LocalDate dateOfBirth): void
+ editUser(String firstName, String lastName, String email, LocalDate dateOfBirth): void
+ getUser(String username, String password): User
+ removeUser(String username, String password): void
CourseList
- courses: ArrayList<Course>
- courseList: CourseList
- CourseList()
+ getInstance(): CourseList
+ addCourse(Language language, ArrayList<Lesson> lessons)
+ getCourse(Language language): Course
DataReadWriter
+ getUsers(): ArrayList<User>
+ updateUsers(ArrayList<User>): void
+ getCourses(): ArrayList<Course>
+ updateCourseProgress(Hashap<Course> Courses): void
UserList JSON
"users": [
{
"uuid"
"firstname"
"lastname"
"email"
"phone number"
"ProfPoints"
"settings": {
"emailNotifications"
"darkMode"
}
"courses": [
{
"language"
"progress"
}
]
}
CourseList JSON
"course": {
"language"
"progress"
"lessons": [
{
"title"
"progress"
"intro"
"keywords": [],
"questions": [
{
"question"
"type"
"options"
"answer"
},
}
}
Settings
- emailNotifications: boolean
- isDarkMode: boolean
- user: User
+ toggleDarkMode(boolean isDarkMode): void
+ toggleEmailNotification(boolean emailNotification): void
User
- username: String
- password: String
- UUID: id
- firstName: String
- lastName: String
- emailAddress: String
- dateOfBirth: LocalDate
- courses: HashMap<Course course, int progress>
- proficiencyPointTotal: int
- settings: Settings
+ User(String firstName, String lastName, String email, LocalDate dateOfBirth)
+ User(String firstName, String lastName, String email)
+ getUsername(): String
+ getFirstname(): String
+ getLastName(): String
+ getEmailAddress(): String
+ getDateOfBirth(): LocalDate
+ getCourseProgress(Course course): int
+ setName(String firstName, String lastName): void
+ setEmailAddress(String emailAddress): void
+ setDateOfBirth(LocalDate dateOfBirth): void
+ setCourseProgress(ArrayList<int> lessonProgress): void
+ addCourseProgress(String language , Int lessonProgress): void
+ toString(): Void
+ isMatch(String username, String password): User
Course
- language: Language
- lessons: ArrayList<Lesson>
- proficiencyPts: int
- UUID: id
+ Course (Language language)
+ getLesson(int Index): Lesson
- addLesson(Lesson lesson): void
+ getLanguage(): Language
+ getProgress(): int
+ toString(): String
<<enumeration>>
Language
SPANISH
FRENCH
JAPANESE
<<enumeration>>
GoalType
LESSONS
PROFICIENCY_POINTS
Goal
- goalType: GoalType
- tracker: int
- isWeekly: bool
+ Goal(GoalType goalType, int goalAmt)
+ metGoal(int tracker): boolean
Lesson
- subject: String
- exercises: ArrayList<Exercise>
- words: Word[]
- phrases: Phrase[]
- progress: int
+ Lesson(String, ArrayList<Excerise>, Word[] words, Phrase[] phrases, int progress)
+ getExercise(Int index): Exercise
- addExercise(Exercise exercise): void
- calcProgress(): void
+ getSubject(): String
+ getProgress(): int
+ toString(): String
+ addProficiencyPoint(); void
Exercise
question: String
options: String
answer: Word
+ Exercise(String question, Word answer)
+ isCorrect(String useranswer): bool
+ toString(): String
Matching
- matchingArray: String[][]
+ Matching(String question, String answer)
+ isCorrect(String useranswer): bool
+ toString(): String
Fill_In
- answers: Word[]
+ Fill_In(String question, String answer)
+ isCorrect(String useranswer): bool
+ toString(): String
Conversation
- answers: Word[]
+ Conversation(String question, String answer)
+ isCorrect(String useranswer): bool
+ toString(): String
Word
- word: String
- meaning: String
- phonetics: String
+ Word(String word, String meaning, String phonetics):void
+ getWord():String
+ getMeaning():String
+ getPhonetics:String
+ toString():String
Phrase
- phrase: Word[]
- translation: String
- meaning: String
+ Phrase(String phrase, String translation, String meaning): void
+ getPhrase(): String
+ getTranslation(): String
+ getMeaning(): String
+ toString(): String