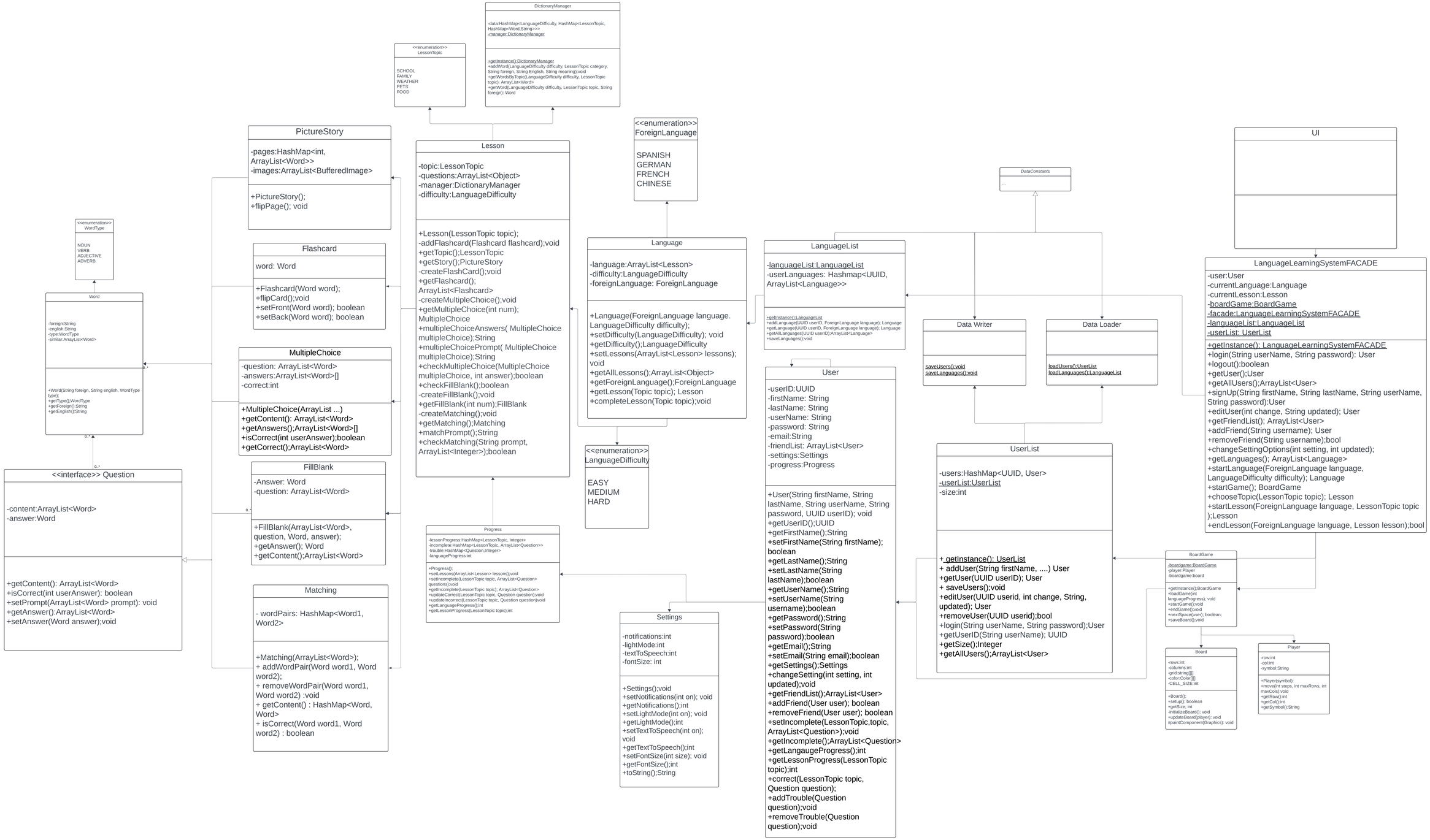
DictionaryManager
-data:HashMap<LanguageDifficulty, HashMap<LessonTopic, HashMap<Word,String>>>
-manager:DictionaryManager
+getInstance():DictionaryManager
+addWord(LanguageDifficulty difficulty, LessonTopic category, String foreign, String English, String meaning):void
+getWordsByTopic(LanguageDifficulty difficulty, LessonTopic topic): ArrayList<Word>
+getWord(LanguageDifficulty difficulty, LessonTopic topic, String foreign): Word
<<enumeration>>
LessonTopic
SCHOOL
FAMILY
WEATHER
PETS
FOOD
PictureStory
-pages:HashMap<int, ArrayList<Word>>
-images:ArrayList<BufferedImage>
+PictureStory();
+flipPage(); void
Lesson
-topic:LessonTopic
-questions:ArrayList<Object>
-manager:DictionaryManager
-difficulty:LanguageDifficulty
+Lesson(LessonTopic topic);
-addFlashcard(Flashcard flashcard);void
+getTopic();LessonTopic
+getStory();PictureStory
-createFlashCard();void
+getFlashcard(); ArrayList<Flashcard>
-createMultipleChoice();void
+getMultipleChoice(int num); MultipleChoice
+multipleChoiceAnswers( MultipleChoice multipleChoice);String
+multipleChoicePrompt( MultipleChoice multipleChoice);String
+checkMultipleChoice(MultipleChoice multipleChoice, int answer);boolean
+checkFillBlank();boolean
-createFillBlank();void
+getFillBlank(int num);FillBlank
-createMatching();void
+getMatching();Matching
+matchPrompt();String
+checkMatching(String prompt, ArrayList<Integer>);boolean
<<enumeration>>
ForeignLanguage
SPANISH
GERMAN
FRENCH
CHINESE
<<enumeration>>
WordType
NOUN
VERB
ADJECTIVE
ADVERB
Word
-foreign:String
-english:String
-type:WordType
-similar:ArrayList<Word>
+Word(String foreign, String english, WordType type);
+getType():WordType
+getForeign():String
+getEnglish():String
Flashcard
word: Word
+Flashcard(Word word);
+flipCard();void
+setFront(Word word); boolean
+setBack(Word word); boolean
MultipleChoice
-question: ArrayList<Word>
-answers:ArrayList<Word>[]
-correct:int
+MultipleChoice(ArrayList ...)
+getContent(): ArrayList<Word>
+getAnswers();ArrayList<Word>[]
+isCorrect(int userAnswer);boolean
+getCorrect();ArrayList<Word>
FillBlank
-Answer: Word
-question: ArrayList<Word>
+FillBlank(ArrayList<Word>, question, Word, answer);
+getAnswer(); Word
+getContent();ArrayList<Word>
Matching
- wordPairs: HashMap<Word1, Word2>
+Matching(ArrayList<Word>);
+ addWordPair(Word word1, Word word2);
+ removeWordPair(Word word1, Word word2) :void
+ getContent() : HashMap<Word, Word>
+ isCorrect(Word word1, Word word2) : boolean
<<interface>> Question
-content:ArrayList<Word>
-answer:Word
+getContent(): ArrayList<Word>
+isCorrect(int userAnswer): boolean
+setPrompt(ArrayList<Word> prompt): void
+getAnswer():ArrayList<Word>
+setAnswer(Word answer);void
Language
-language:ArrayList<Lesson>
-difficulty:LanguageDifficulty
-foreignLanguage: ForeignLanguage
+Language(ForeignLanguage language. LanguageDifficulty difficulty);
+setDifficulty(LanguageDifficulty); void
+getDifficulty();LanguageDifficulty
+setLessons(ArrayList<Lesson> lessons); void
+getAllLessons();ArrayList<Object>
+getForeignLanguage();ForeignLanguage
+getLesson(Topic topic); Lesson
+completeLesson(Topic topic);void
<<enumeration>>
LanguageDifficulty
EASY
MEDIUM
HARD
Progress
-lessonProgress:HashMap<LessonTopic, Integer>
-incomplete:HashMap<LessonTopic, ArrayList<Question>>
-trouble:HashMap<Question,Integer>
-languageProgress:int
+Progress();
+setLessons(ArrayList<Lesson> lessons);void
+setIncomplete(LessonTopic topic, ArrayList<Question> questions);void
+getIncomplete(LessonTopic topic); ArrayList<Question>
+updateCorrect(LessonTopic topic, Question question);void
+updateIncorrect(LessonTopic topic, Question question)void
+getLanguageProgress();int
+getLessonProgress(LessonTopic topic);int
Settings
-notifications:int
-lightMode:int
-textToSpeech:int
-fontSize: int
+Settings();void
+setNotifications(int on); void
+getNotifications();int
+setLightMode(int on); void
+getLightMode();int
+setTextToSpeech(int on); void
+getTextToSpeech();int
+setFontSize(int size); void
+getFontSize();int
+toString();String
LanguageList
-languageList:LanguageList
-userLanguages: Hashmap<UUID, ArrayList<Language>>
+getInstance():LanguageList
+addLanguage(UUID userID, ForeignLanguage language); Language
+getLanguage(UUID userID, ForeignLanguage language); Language
+getAllLanguages(UUID userID):ArrayList<Language>
+saveLanguages();void
User
-userID:UUID
-firstName: String
-lastName: String
-userName: String
-password: String
-email:String
-friendList: ArrayList<User>
-settings:Settings
-progress:Progress
+User(String firstName, String lastName, String userName, String password, UUID userID); void
+getUserID();UUID
+getFirstName();String
+setFirstName(String firstName); boolean
+getLastName();String
+setLastName(String lastName);boolean
+getUserName();String
+setUserName(String username);boolean
+getPassword();String
+setPassword(String password);boolean
+getEmail();String
+setEmail(String email);boolean
+getSettings();Settings
+changeSetting(int setting, int updated);void
+getFriendList();ArrayList<User>
+addFriend(User user); boolean
+removeFriend(User user); boolean
+setIncomplete(LessonTopic,topic, ArrayList<Question>);void
+getIncomplete();ArrayList<Question>
+getLangaugeProgress();int
+getLessonProgress(LessonTopic topic);int
+correct(LessonTopic topic, Question question);
+addTrouble(Question question);void
+removeTrouble(Question question);void
DataConstants
...
Data Writer
saveUsers();void
saveLanguages();void
Data Loader
loadUsers();UserList
loadLanguages();LanguageList
UserList
-users:HashMap<UUID, User>
-userList:UserList
-size:int
+ getInstance(): UserList
+ addUser(String firstName, ....) User
+getUser(UUID userID); User
+ saveUsers();void
+editUser(UUID userid, int change, String, updated); User
+removeUser(UUID userid);bool
+login(String userName, String password);User
+getUserID(String userName); UUID
+getSize();Integer
+getAllUsers();ArrayList<User>
UI
LanguageLearningSystemFACADE
-user:User
-currentLanguage:Language
-currentLesson:Lesson
-boardGame:BoardGame
-facade:LanguageLearningSystemFACADE
-languageList:LanguageList
-userList: UserList
+getInstance(): LanguageLearningSystemFACADE
+login(String userName, String password): User
+logout():boolean
+getUser();User
+getAllUsers();ArrayList<User>
+signUp(String firstName, String lastName, String userName, String password):User
+editUser(int change, String updated); User
+getFriendList(); ArrayList<User>
+addFriend(String username); User
+removeFriend(String username);bool
+changeSettingOptions(int setting, int updated);
+getLanguages(); ArrayList<Language>
+startLanguage(ForeignLanguage language, LanguageDifficulty difficulty); Language
+startGame(); BoardGame
+chooseTopic(LessonTopic topic); Lesson
+startLesson(ForeignLanguage language, LessonTopic topic );Lesson
+endLesson(ForeignLanguage language, Lesson lesson);bool
BoardGame
-boardgame:BoardGame
-player:Player
-boardgame:board
+getInstance();BoardGame
+loadGame(int languageProgress); void
+startGame();void
+endGame();void
+nextSpace(user); boolean;
+saveBoard();void
Board
-rows:int
-columns:int
-grid:string[][]
-color:Color[][]
-CELL_SIZE:int
+Board();
+setup(); boolean
+getSize; int
-initializeBoard(): void
+updateBoard(player): void
#paintComponent(Graphics): void
Player
-row:int
-col:int
-symbol:String
+Player(symbol);
+move(int steps, int maxRows, int maxCols);void
+getRow();int
+getCol();int
+getSymbol():String