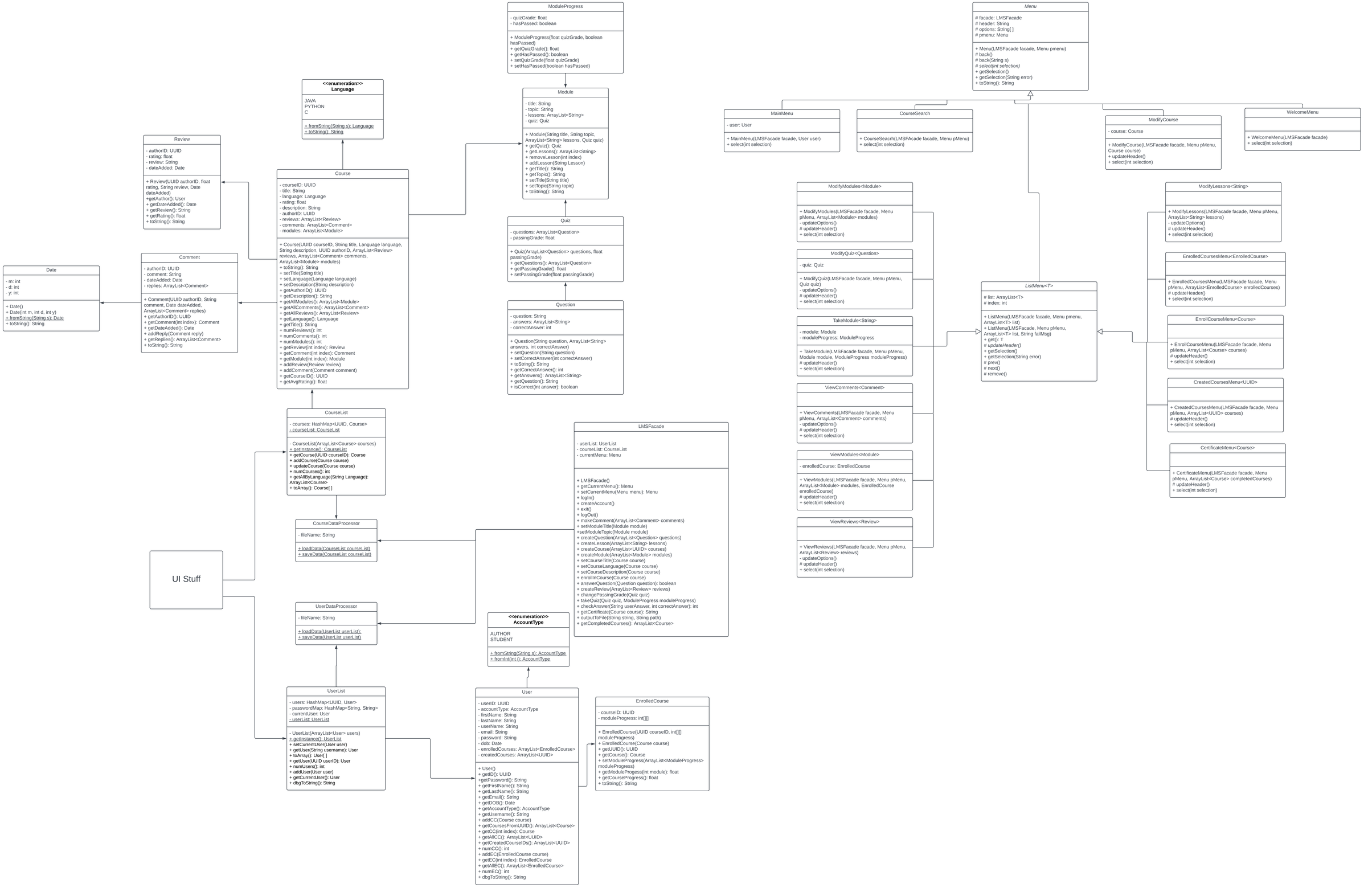
ModuleProgress
- quizGrade: float
- hasPassed: boolean
+ ModuleProgress(float quizGrade, boolean hasPassed)
+ getQuizGrade(): float
+ getHasPassed(): boolean
+ setQuizGrade(float quizGrade)
+ setHasPassed(boolean hasPassed)
<<enumeration>>
Language
JAVA
PYTHON
C
+ fromString(String s): Language
+ toString(): String
Module
- title: String
- topic: String
- lessons: ArrayList<String>
- quiz: Quiz
+ Module(String title, String topic, ArrayList<String> lessons, Quiz quiz)
+ getQuiz(): Quiz
+ getLessons(): ArrayList<String>
+ removeLesson(int index)
+ addLesson(String Lesson)
+ getTitle(): String
+ getTopic(): String
+ setTitle(String title)
+ setTopic(String topic)
+ toString(): String
Review
- authorID: UUID
- rating: float
- review: String
- dateAdded: Date
+ Review(UUID authorID, float rating, String review, Date dateAdded)
+getAuthor(): User
+ getDateAdded(): Date
+ getReview(): String
+ getRating(): float
+ toString(): String
Course
- courseID: UUID
- title: String
- language: Language
- rating: float
- description: String
- authorID: UUID
- reviews: ArrayList<Review>
- comments: ArrayList<Comment>
- modules: ArrayList<Module>
+ Course(UUID courseID, String title, Language language, String description, UUID authorID, ArrayList<Review> reviews, ArrayList<Comment> comments, ArrayList<Module> modules)
+ toString(): String
+ setTitle(String title)
+ setLanguage(Language language)
+ setDescription(String description)
+ getAuthorID(): UUID
+ getDescription(): String
+ getAllModules(): ArrayList<Module>
+ getAllComments(): ArrayList<Comment>
+ getAllReviews(): ArrayList<Review>
+ getLanguage(): Language
+ getTitle(): String
+ numReviews(): int
+ numComments(): int
+ numModules(): int
+ getReview(int index): Review
+ getComment(int index): Comment
+ getModule(int index): Module
+ addReview(Review review)
+ addComment(Comment comment)
+ getCourseID(): UUID
+ getAvgRating(): float
Quiz
- questions: ArrayList<Question>
- passingGrade: float
+ Quiz(ArrayList<Question> questions, float passingGrade)
+ getQuestions(): ArrayList<Question>
+ getPassingGrade(): float
+ setPassingGrade(float passingGrade)
Date
- m: int
- d: int
- y: int
+ Date()
+ Date(int m, int d, int y)
+ fromString(String s): Date
+ toString(): String
Comment
- authorID: UUID
- comment: String
- dateAdded: Date
- replies: ArrayList<Comment>
+ Comment(UUID authorID, String comment, Date dateAdded, ArrayList<Comment> replies)
+ getAuthorID(): UUID
+ getComment(int index): Comment
+ getDateAdded(): Date
+ addReply(Comment reply)
+ getReplies(): ArrayList<Comment>
+ toString(): String
Question
- question: String
- answers: ArrayList<String>
- correctAnswer: int
+ Question(String question, ArrayList<String> answers, int correctAnswer)
+ setQuestion(String question)
+ setCorrectAnswer(int correctAnswer)
+ toString(): String
+ getCorrectAnswer(): int
+ getAnswers(): ArrayList<String>
+ getQuestion(): String
+ isCorrect(int answer): boolean
CourseList
- courses: HashMap<UUID, Course>
- courseList: CourseList
- CourseList(ArrayList<Course> courses)
+ getInstance(): CourseList
+ getCourse(UUID courseID): Course
+ addCourse(Course course)
+ updateCourse(Course course)
+ numCourses(): int
+ getAllByLanguage(String Language): ArrayList<Course>
+ toArray(): Course[ ]
LMSFacade
- userList: UserList
- courseList: CourseList
- currentMenu: Menu
+ LMSFacade()
+ getCurrentMenu(): Menu
+ setCurrentMenu(Menu menu): Menu
+ login()
+ createAccount()
+ exit()
+ logOut()
+ makeComment(ArrayList<Comment> comments)
+ setModuleTitle(Module module)
+setModuleTopic(Module module)
+ createQuestion(ArrayList<Question> questions)
+ createLesson(ArrayList<String> lessons)
+ createCourse(ArrayList<UUID> courses)
+ createModule(ArrayList<Module> modules)
+ setCourseTitle(Course course)
+ setCourseLanguage(Course course)
+ setCourseDescription(Course course)
+ enrollInCourse(Course course)
+ answerQuestion(Question question): boolean
+ createReview(ArrayList<Review> reviews)
+ changePassingGrade(Quiz quiz)
+ takeQuiz(Quiz quiz, ModuleProgress moduleProgress)
+ checkAnswer(String userAnswer, int correctAnswer): int
+ getCertificate(Course course): String
+ outputToFile(String string, String path)
+ getCompletedCourses(): ArrayList<Course>
CourseDataProcessor
- fileName: String
+ loadData(CourseList courseList)
+ saveData(CourseList courseList)
UI Stuff
UserDataProcessor
- fileName: String
+ loadData(UserList userList):
+ saveData(UserList userList)
<<enumeration>>
AccountType
AUTHOR
STUDENT
+ fromString(String s): AccountType
+ fromInt(int i): AccountType
UserList
- users: HashMap<UUID, User>
- passwordMap: HashMap<String, String>
- currentUser: User
- userList: UserList
- UserList(ArrayList<User> users)
+ getInstance(): UserList
+ setCurrentUser(User user)
+ getUser(String username): User
+ toArray(): User[ ]
+ getUser(UUID userID): User
+ numUsers(): int
+ addUser(User user)
+ getCurrentUser(): User
+ dbgToString(): String
User
- userID: UUID
- accountType: AccountType
- firstName: String
- lastName: String
- userName: String
- email: String
- password: String
- dob: Date
- enrolledCourses: ArrayList<EnrolledCourse>
- createdCourses: ArrayList<UUID>
+ User()
+ getID(): UUID
+getPassword(): String
+ getFirstName(): String
+ getLastName(): String
+ getEmail(): String
+ getDOB(): Date
+ getAccountType(): AccountType
+ getUsername(): String
+ addCC(Course course)
+ getCoursesFromUUID(): ArrayList<Course>
+ getCC(int index): Course
+ getAllCC(): ArrayList<UUID>
+ getCreatedCourseIDs(): ArrayList<UUID>
+ numCC(): int
+ addEC(EnrolledCourse course)
+ getEC(int index): EnrolledCourse
+ getAllEC(): ArrayList<EnrolledCourse>
+ numEC(): int
+ dbgToString(): String
EnrolledCourse
- courseID: UUID
- moduleProgress: int[][]
+ EnrolledCourse(UUID courseID, int[][] moduleProgress)
+ EnrolledCourse(Course course)
+ getUUID(): UUID
+ getCourse(): Course
+ setModuleProgress(ArrayList<ModuleProgress> moduleProgress)
+ getModuleProgess(int module): float
+ getCourseProgress(): float
+ toString(): String
Menu
facade: LMSFacade
header: String
options: String[ ]
pmenu: Menu
+ Menu(LMSFacade facade, Menu pmenu)
# back()
# back(String s)
# select(int selection)
+ getSelection()
+ getSelection(String error)
+ toString(): String
MainMenu
- user: User
+ MainMenu(LMSFacade facade, User user)
+ select(int selection)
CourseSearch
+ CourseSeacrh(LMSFacade facade, Menu pMenu)
+ select(int selection)
ModifyCourse
- course: Course
+ ModifyCourse(LMSFacade facade, Menu pMenu, Course course)
+ updateHeader()
+ select(int selection)
WelcomeMenu
+ WelcomeMenu(LMSFacade facade)
+ select(int selection)
ModifyModules<Module>
+ ModifyModules(LMSFacade facade, Menu pMenu, ArrayList<Module> modules)
- updateOptions()
updateHeader()
+ select(int selection)
ModifyQuiz<Question>
- quiz: Quiz
+ ModifyQuiz(LMSFacade facade, Menu pMenu, Quiz quiz)
- updateOptions()
updateHeader()
+ select(int selection)
TakeModule<String>
- module: Module
- moduleProgress: ModuleProgress
+ TakeModule(LMSFacade facade, Menu pMenu, Module module, ModuleProgress moduleProgress)
# updateHeader()
+ select(int selection)
ViewComments<Comment>
+ ViewComments(LMSFacade facade, Menu pMenu, ArrayList<Comment> comments)
- updateOptions()
updateHeader()
+ select(int selection)
ViewModules<Module>
- enrolledCourse: EnrolledCourse
+ ViewModules(LMSFacade facade, Menu pMenu, ArrayList<Module> modules, EnrolledCourse enrolledCourse)
# updateHeader()
+ select(int selection)
ViewReviews<Review>
+ ViewReviews(LMSFacade facade, Menu pMenu, ArrayList<Review> reviews)
- updateOptions()
updateHeader()
+ select(int selection)
ListMenu<T>
list: ArrayList<T>
index: int
+ ListMenu(LMSFacade facade, Menu pmenu, ArrayList<T> list)
+ ListMenu(LMSFacade, Menu pMenu, ArrayList<T> list, String failMsg)
+ get(): T
# updateHeader()
+ getSelection()
+ getSelection(String error)
# prev()
# next()
# remove()
ModifyLessons<String>
+ ModifyLessons(LMSFacade facade, Menu pMenu, ArrayList<String> lessons)
- updateOptions()
updateHeader()
+ select(int selection)
EnrolledCoursesMenu<EnrolledCourse>
+ EnrolledCoursesMenu(LMSFacade facade, Menu pMenu, ArrayList<EnrolledCourse> enrolledCourses)
# updateHeader()
+ select(int selection)
EnrollCourseMenu<Course>
+ EnrollCourseMenu(LMSFacade facade, Menu pMenu, ArrayList<Course> courses)
# updateHeader()
+ select(int selection)
CreatedCoursesMenu<UUID>
+ CreatedCoursesMenu(LMSFacade facade, Menu pMenu, ArrayList<UUID> courses)
# updateHeader()
+ select(int selection)
CertificateMenu<Course>
+ CertificateMenu(LMSFacade facade, Menu pMenu, ArrayList<Course> completedCourses)
# updateHeader()
+ select(int selection)