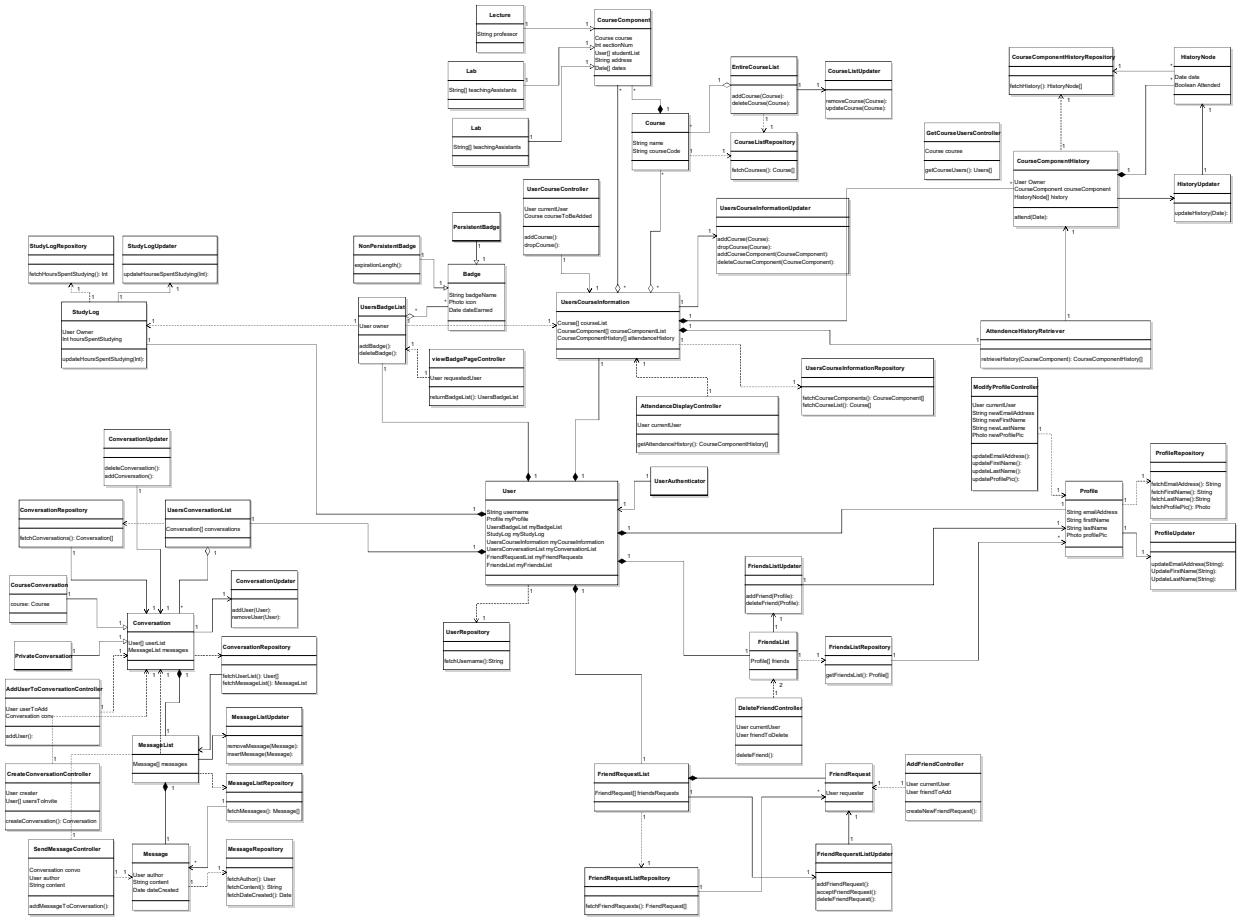
Lecture
String professor
CourseComponent
Course course
Int sectionNum
User[] studentList
String address
Date[] dates
Lab
String[] teachingAssistants
Lab
String[] teachingAssistants
Course
String name
String courseCode
EntireCoursesList
addCourse(Course):
deleteCourse(Course):
CourseListUpdater
removeCourse(Course):
updateCourse(Course):
CourseListRepository
fetchCourses(): Course[]
CourseComponentHistoryRepository
fetchHistory(): HistoryNode[]
HistoryNode
Date date
Boolean Attended
GetCourseUsersController
Course course
getCourseUsers(): Users[]
CourseComponentHistory
User Owner
CourseComponent courseComponent
HistoryNode[] history
attend(Date):
HistoryUpdater
updateHistory(Date):
UserCourseController
User currentUser
Course courseToBeAdded
addCourse():
dropCourse():
UsersCourseInformationUpdater
addCourse(Course):
dropCourse(Course):
addCourseComponent(CourseComponent):
deleteCourseComponent(CourseComponent):
PersistentBadge
NonPersistentBadge
expirationLength():
Badge
String badgeName
Photo icon
Date dateEarned
StudyLogRepository
fetchHoursSpentStudying(): Int
StudyLogUpdater
updateHourseSpentStudying(Int):
StudyLog
User Owner
Int hoursSpentStudying
updateHoursSpentStudying(Int):
UsersBadgeList
User owner
addBadge():
deleteBadge():
viewBadgePageController
User requestedUser
returnBadgeList(): UsersBadgeList
UsersCourseInformation
Course[] courseList
CourseComponent[] courseComponentList
CourseComponentHistory[][] attendanceHistory
AttendanceHistoryRetriever
retrieveHistory(CourseComponent): CourseComponentHistory[]
UsersCourseInformationRepository
fetchCourseComponents(): CourseComponent[]
fetchCourseList(): Course[]
AttendanceDisplayController
User currentUser
getAttendanceHistory(): CourseComponentHistory[]
ModifyProfileController
User currentUser
String newEmailAddress
String newFirstName
String newLastName
Photo newProfilePic
updateEmailAddress():
updateFirstName():
updateLastName():
updateProfilePic():
ProfileRepository
fetchEmailAddress(): String
fetchFirstName(): String
fetchLastName(): String
fetchProfilePic(): Photo
ConversationUpdater
deleteConversation():
addConversation():
ConversationRepository
fetchConversations(): Conversation[]
UsersConversationList
Conversation[] conversations
User
String username
Profile myProfile
UsersBadgeList myBadgeList
StudyLog myStudyLog
UsersCourseInformation myCourseInformation
UsersConversationList myConversationList
FriendRequestList myFriendRequests
FriendsList myFriendsList
UserAuthenticator
Profile
String emailAddress
String firstName
String lastName
Photo profilePic
ProfileUpdater
updateEmailAddress(String):
UpdateFirstName(String):
UpdateLastName(String):
CourseConversation
course: Course
ConversationUpdater
addUser(User):
removeUser(User):
Conversation
User[] userList
MessageList messages
PrivateConversation
ConversationRepository
fetchUserList(): User[]
fetchMessageList(): MessageList
UserRepository
fetchUsername(): String
FriendsListUpdater
addFriend(Profile):
deleteFriend(Profile):
FriendsList
Profile[] friends
FriendsListRepository
getFriendsList(): Profile[]
AddUserToConversationController
User userToAdd
Conversation conv
addUser():
MessageListUpdater
removeMessage(Message):
insertMessage(Message):
MessageList
Message[] messages
DeleteFriendController
User currentUser
User friendToDelete
deleteFriend():
CreateConversationController
User creater
User[] usersToInvite
createConversation(): Conversation
MessageListRepository
fetchMessages(): Message[]
FriendRequestList
FriendRequest[] friendsRequests
FriendRequest
User requester
AddFriendController
User currentUser
User friendToAdd
createNewFriendRequest():
SendMessageController
Conversation convo
User author
String content
addMessageToConversation():
Message
User author
String content
Date dateCreated
MessageRepository
fetchAuthor(): User
fetchContent(): String
fetchDateCreated(): Date
FriendRequestsListUpdater
addFriendRequest():
acceptFriendRequest():
deleteFriendRequest():
FriendRequestsListRepository
fetchFriendRequests(): FriendRequest[]