ProfileList Profile profileList : ArrayList<Profile> # username : String + addProfile(profile : Profile) ArrayAdapter + getUsername() : String + getList(): ArrayList<Profile> + setUsername(username : String) + removeProfile(profile : Profile) + setList(profileList : ArrayList<Profile>) + clearList() UserProfile extends - following : FollowerList - followers : FollowerList - requestInbox : FollowRequestInbox HabitListAdapter FollowerList ArrayList<Habit> -habits: HabitList + followUser(profile : UserProfile) -context: Context + unfollowUser(profile : UserProfile) + addFollower(profile : UserProfile) + removeFollower(profile : UserProfile) +getView(): View + sendFollowRequest(target : UserProfile) + recieveFollowRequest(sender : UserProfile) + getFollowers(profile: UserProfile) +getFollowing(profile: UserProfile) 0...* <<entity>> FollowRequestInbox title <= 20 char Habit FollowRequest reason <= 30 char - requests : Arraylist<FollowRequest> dateToStart: yyyy-MM-dd -title: String - sender : UserProfile format - owner : UserProfile -reason: String - target : UserProfile -hid: UUID -dateToStart: String + addRequest(request : FollowRequest) -publicVisibility: Boolean + getSender(): UserProfile + removeRequest(request : FollowRequest) + acceptRequest(request : FollowRequest) : FollowRequest +weeklySchedule WeekSchedule + setSender(profile : UserProfile) +completionSchedule: + getTarget() : UserProfile + denyRequest (request : FollowRequest) CompletionSchedule + setTarget(profile : UserProfile) + setOwner(profille : UserProfile) + getOwner(): UserProfile -titleConstraint(title: String): Boolean -reasonConstraint(reason: String): Boolean WeeklySchedule -schedule: Hashtable<String, Boolean> +clearTable(): void +fillTable(): void +getSchedule(): ArrayList<String> Hashtable has Key: weekday and Value: boolean to represent if the habit occurs on that day or not

Activity UML

