@startuml  
interface Notification {  
 + sendNotification(message: String): void  
}  
  
class NotificationService {  
 + sendNotification(message: String): void  
}  
  
  
class User {  
 + id: UUID  
 + name: String  
 + email: String  
 + profilePhoto: Photo  
 + participateInWalk(walk: Walk): void  
 + orderPhotoAlbum(selectedPhotos: List<Photo>): PhotoAlbum  
 + getParticipatedWalks(): List<Walk>  
 + createdWalks: List<Walk>  
 + askNotifications(walk: Walk) : void  
}  
  
class Ad {   
+ id: UUID  
+ createdAt : Date  
+ updatedAt : Date  
+ title: String  
+ description: String  
+ location : <Location>  
+ creator: <User>  
+ mainPhoto: <Photo>  
+ type : String  
+ feedback : List<Feedback>  
}   
  
class Location {  
 + id: UUID  
 + name: String  
 + address: String  
 + latitude: float  
 + longitude: float  
}  
  
class Walk {  
 + id: UUID  
 + date: Date  
 + participants: List<User>  
 + chat: Chat  
 + maxParticipants: int  
 + addParticipant(user: User): void  
 + isFull(): boolean  
}  
  
class Gallery {  
 + id: UUID  
 + photos: List<Photo>  
 + addPhoto(photo: Photo): void  
}  
  
class Photo {  
 + id: UUID  
 + url: String  
 + description: String  
}  
  
class Feedback {  
 + id: UUID  
 + text: String  
 + gallery: Gallery  
 + user: User  
 + addPhoto(photo: Photo): void  
}  
  
class Chat {  
 + id: UUID  
 + messages: List<Message>  
 + addMessage(message: Message): void  
}  
  
class Message {  
 + id: UUID  
 + sender: <User>  
 + content: String  
 + timestamp: DateTime  
}  
  
class PhotoAlbum {  
 + id: UUID  
 + user: User  
 + selectedPhotos: List<Photo>  
 + generateAlbum(): void  
}  
  
Walk --> Ad : has  
Ad --> Location : located at  
' Notification is triggered whenever Walk is updated  
Walk --> NotificationService : triggers notification on update  
Chat --> NotificationService : triggers notification when new message  
Feedback --> NotificationService : triggers notification when new feedback  
NotificationService --> Notification : implements  
Notification --> User : sent to  
User --> Walk : participates in  
User --> Ad : creates  
User --> Photo : has profile  
User --> Location : resides at  
Feedback --> Gallery : contains  
Gallery --> Photo : has  
Walk --> Photo : has  
Walk --> Chat : associated with  
Ad --> Feedback : receives  
Feedback --> User : posted by  
Chat --> Message : contains  
Message --> User : sender  
PhotoAlbum --> Photo : includes  
User --> PhotoAlbum : can order  
Walk --> User : has participants  
@enduml