@startuml

interface Notification {

+ sendNotification(message: String): void

}

class NotificationService {

+ send(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>

}

class Walk {

+ id: UUID

+ title: String

+ description: String

+ date: Date

+ participants: List<User>

+ gallery: Gallery

+ chat: Chat

+ feedbacks: List<Feedback>

+ creator: User

+ 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

+ message: String

+ photos: List<Photo>

+ 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 --> NotificationService : triggers notification when update

Chat --> NotificationService : triggers notification when new message

Feedback --> NotificationService : triggers notification when new feedback

NotificationService --> Notification :send

Notification --> User : sent to

User --> Walk : participates in

User --> Walk : creates

User --> Photo : has profile

Walk --> Gallery : contains

Gallery --> Photo : has

Walk --> Chat : associated with

Walk --> Feedback : receives

Feedback --> Photo : includes

Feedback --> User : posted by

Chat --> Message : contains

Message --> User : sender

PhotoAlbum --> Photo : includes

User --> PhotoAlbum : can order

Walk --> User : has participants

@enduml