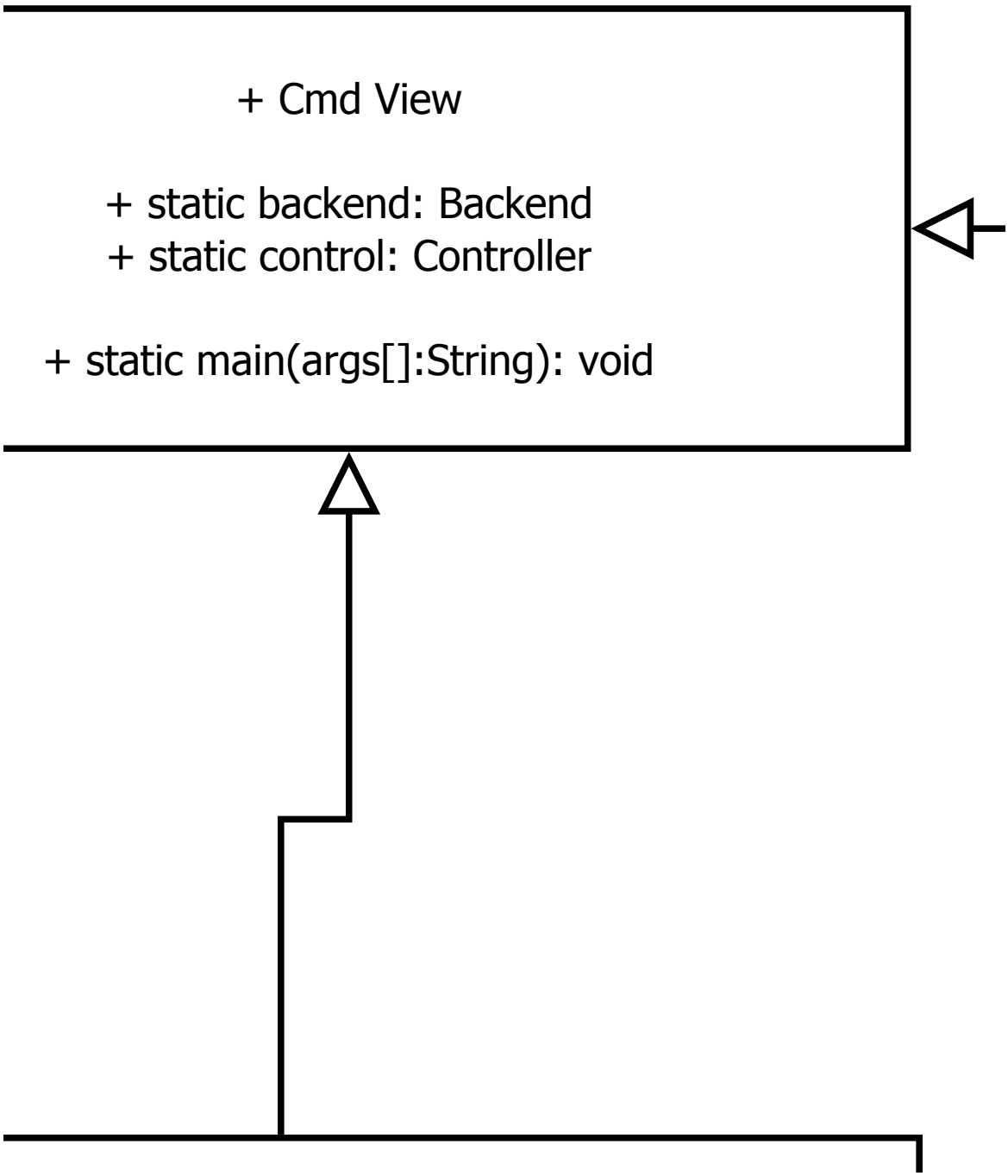
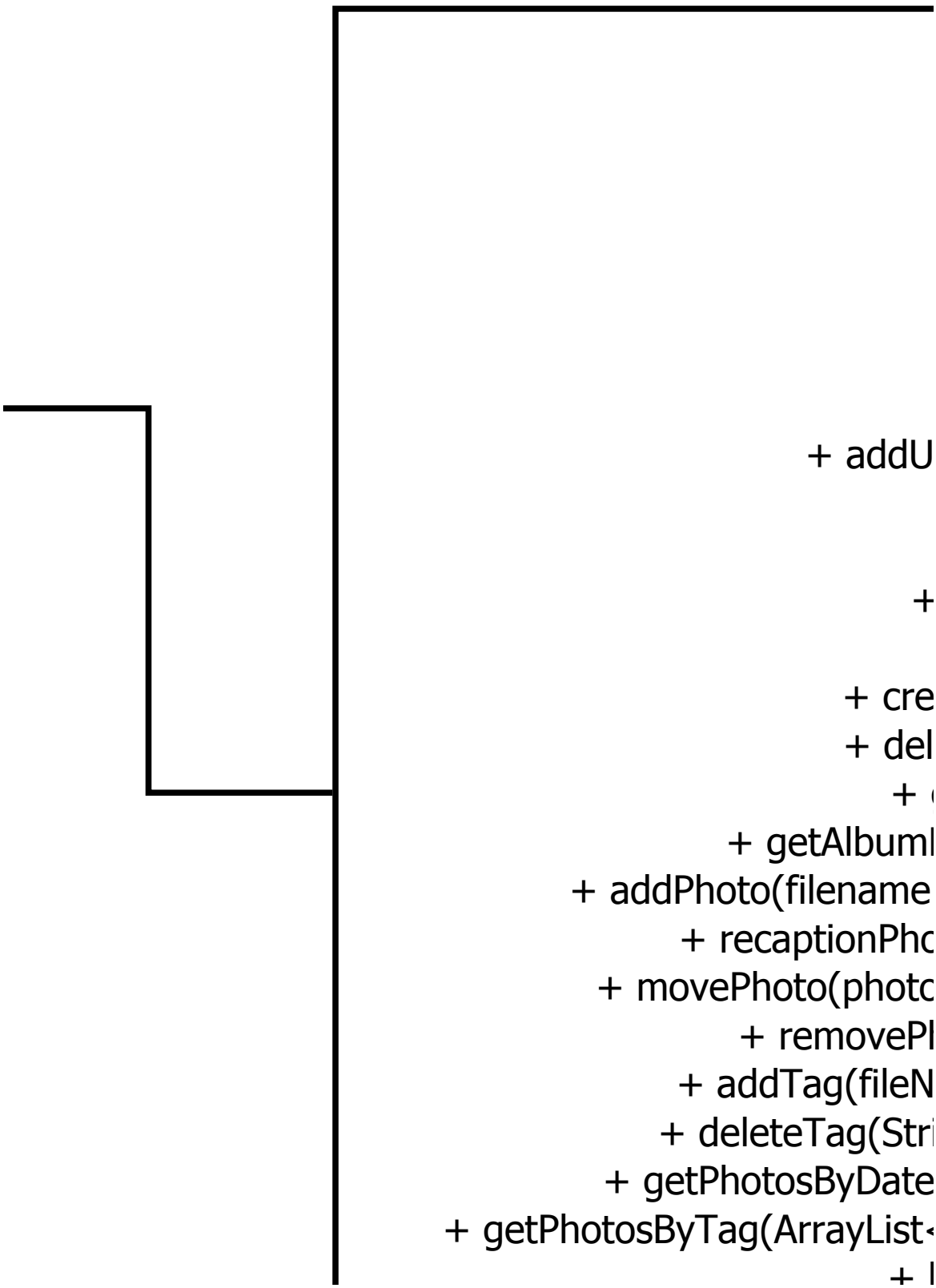
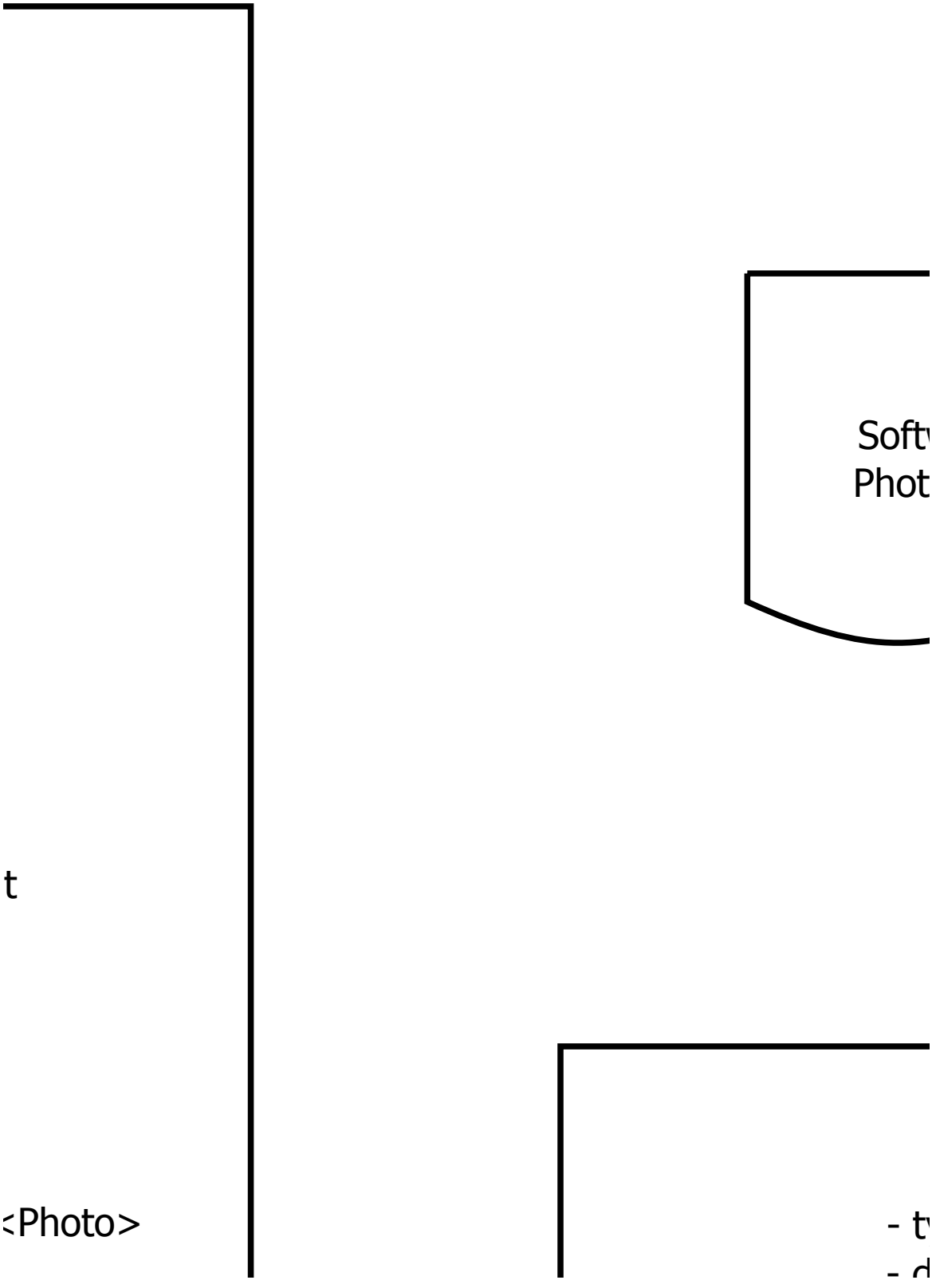

+ Cmd View
+ static backend: Backend
+ static control: Controller
+ static main(args[]:String): void

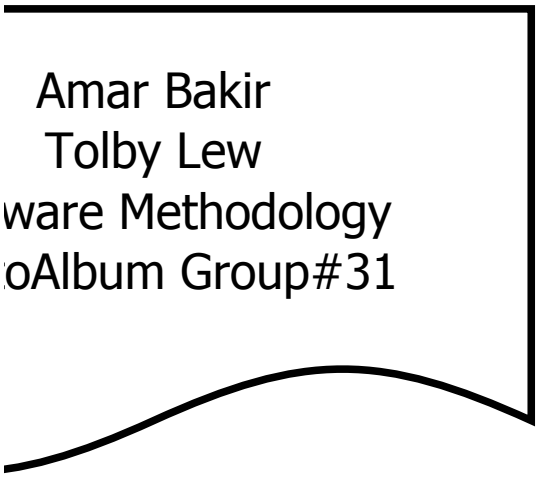

+ addU
+
+ cre
+ del
+
+ getAlbuml
+ addPhoto(filename
+ recaptionPhc
+ movePhoto(photc
+ removePl
+ addTag(fileN
+ deleteTag(Stri
+ getPhotosByDate
+ getPhotosByTag(ArrayList<
+ l

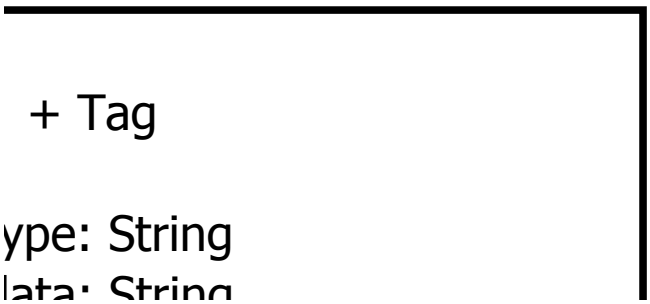
<<interface>>
+ ControlInterface

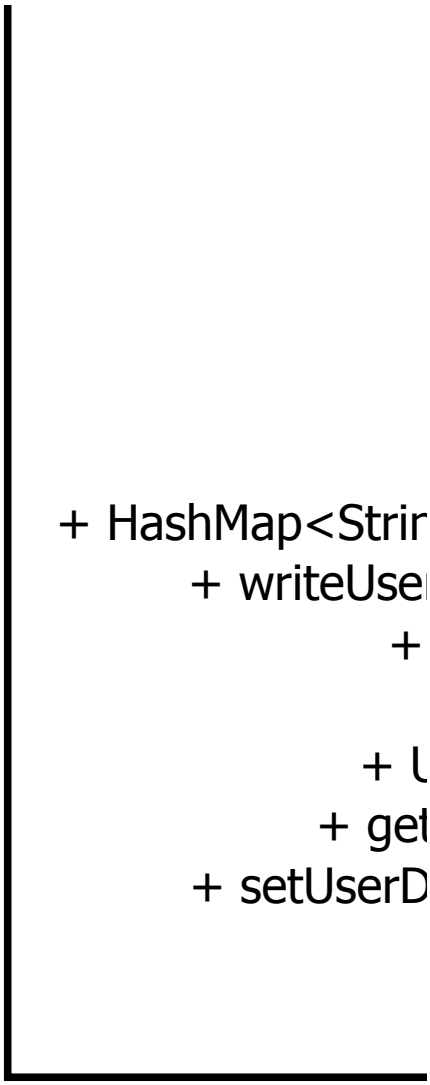
- currentUser: User

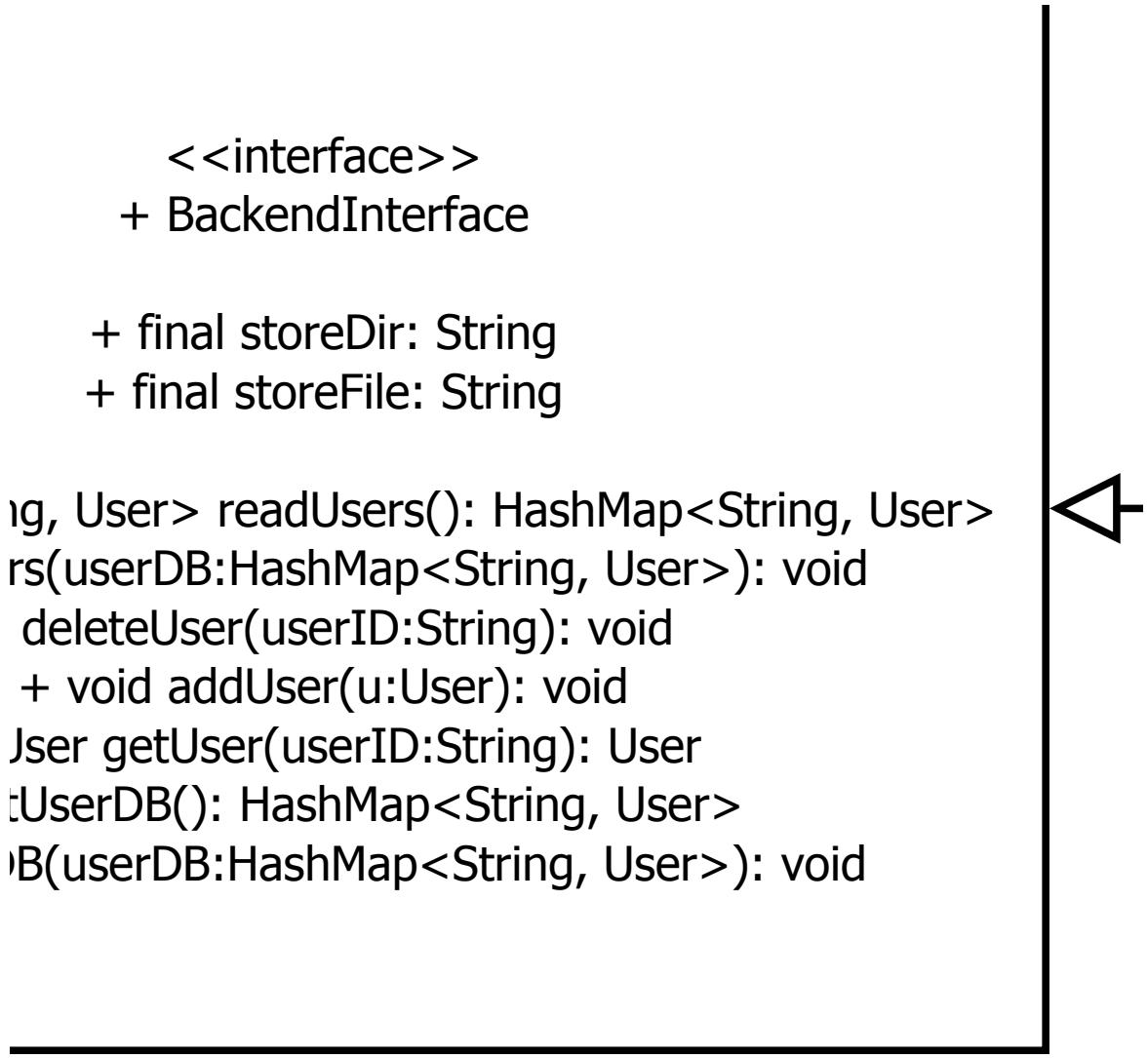
+ listUsers(): ArrayList<User>
ser(ID:String, userName:String): boolean
+ deleteUser(ID:String): boolean
+ login(userID:String): void
· setCurrentUser(User userID): void
+ getCurrentUser(): User
ateAlbum(albumName:String): boolean
eteAlbum(albumName:String): boolean
getUserAlbums(): ArrayList<Album>
Photos(albumName:String): ArrayList<Photo>
:String, caption:String, albumName:String, pic:File): in
oto(fileName:String, newCaption:String): boolean
Name:String, oldAlbum:String, newAlbum:String): int
hoto(fileName:String, albumName:String): int
ame:String, tagType:String, tagValue:String): int
ing fileName, String tagType, String tagValue): int
(String startDate, String endDate): ArrayList<Photo>
<String> keys, ArrayList<String> tagByVal): ArrayList<
listPhotoInfo(String fileName): Photo

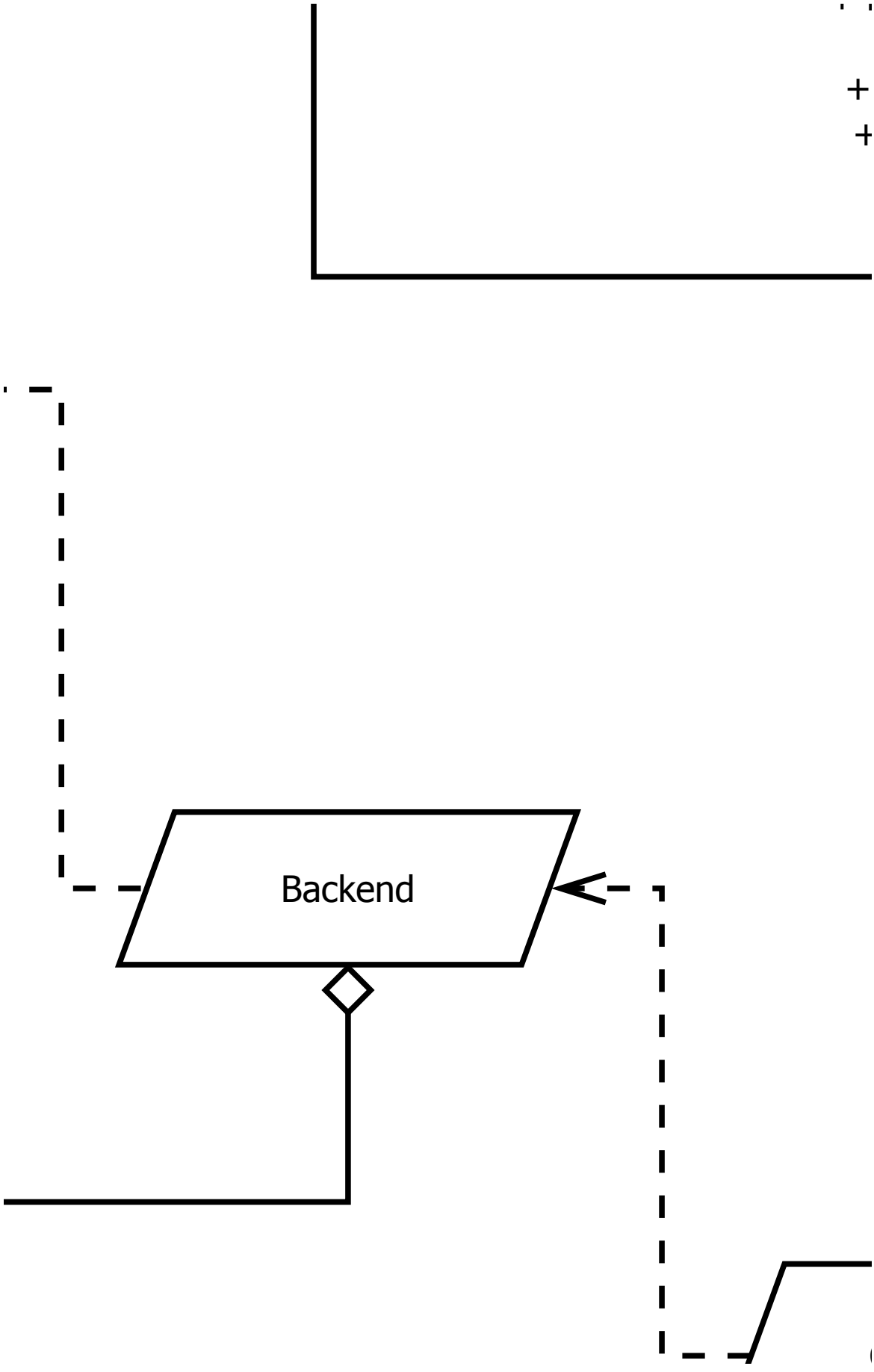

Soft
Phot
t
<Photo>
- t
- d


Amar Bakir
Tolby Lew
ware Methodology
oAlbum Group#31

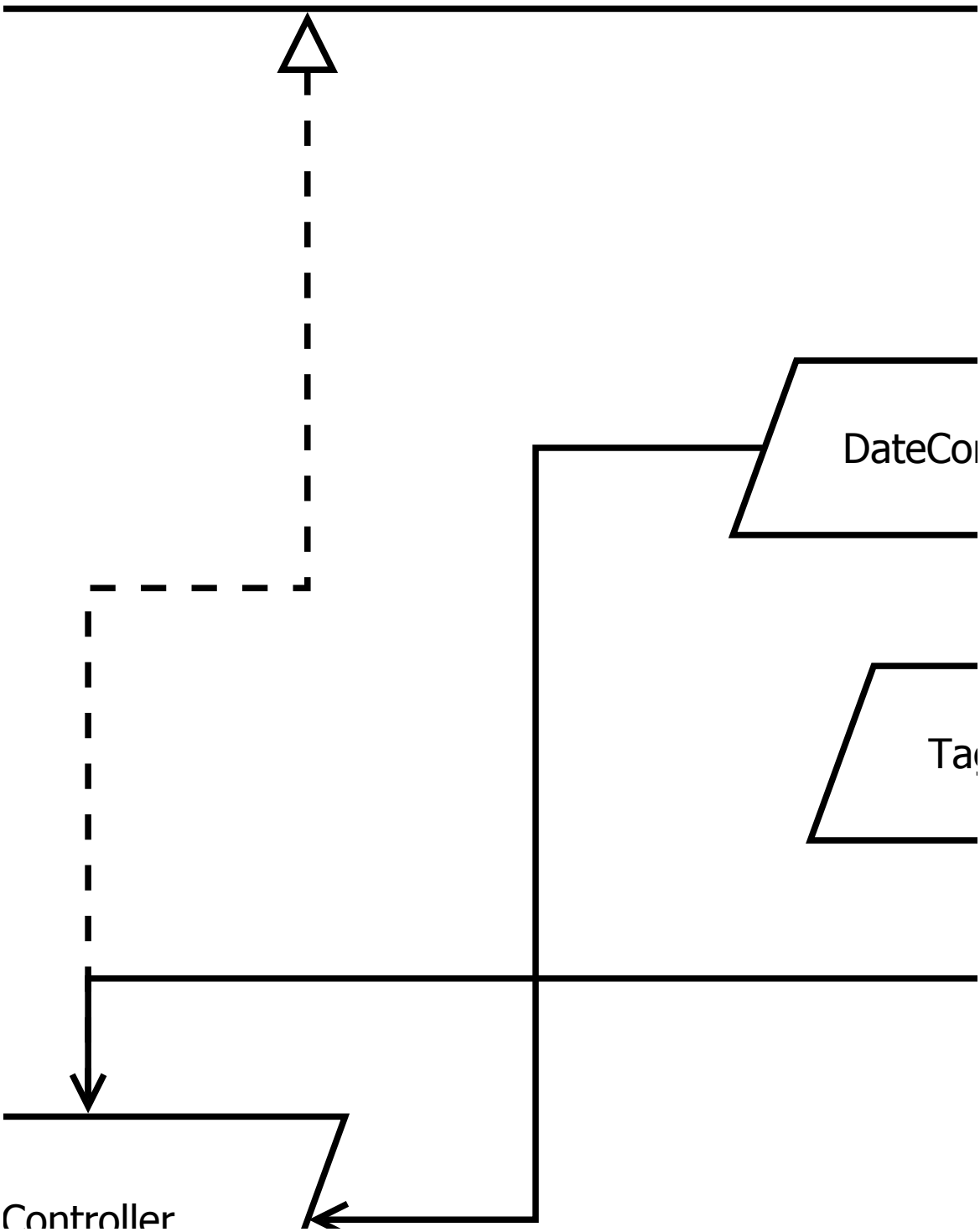

+ Tag
ype: String
lata: String

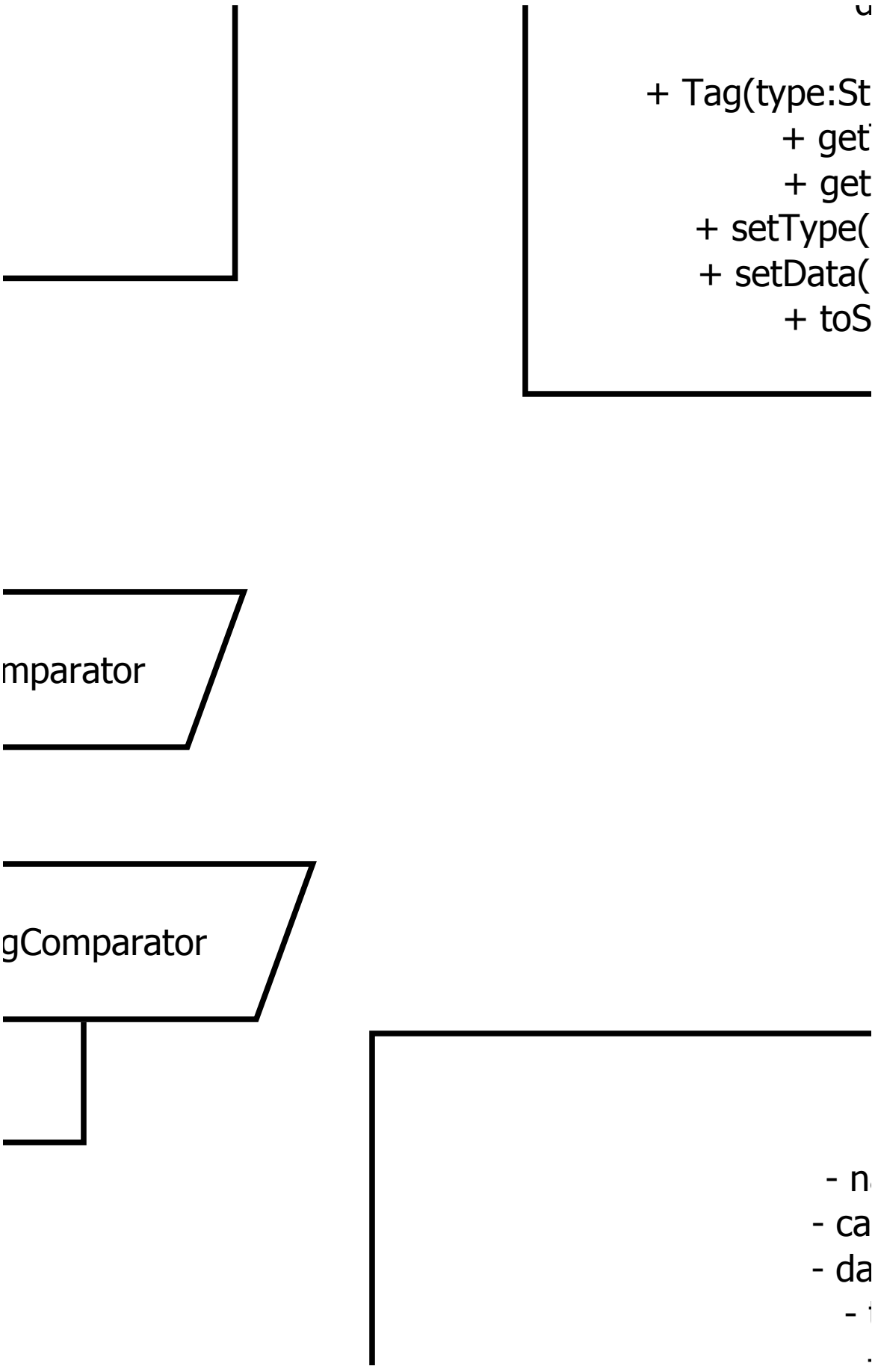

+ HashMap<Strir
+ writeUse
+
+ U
+ get
+ setUserD

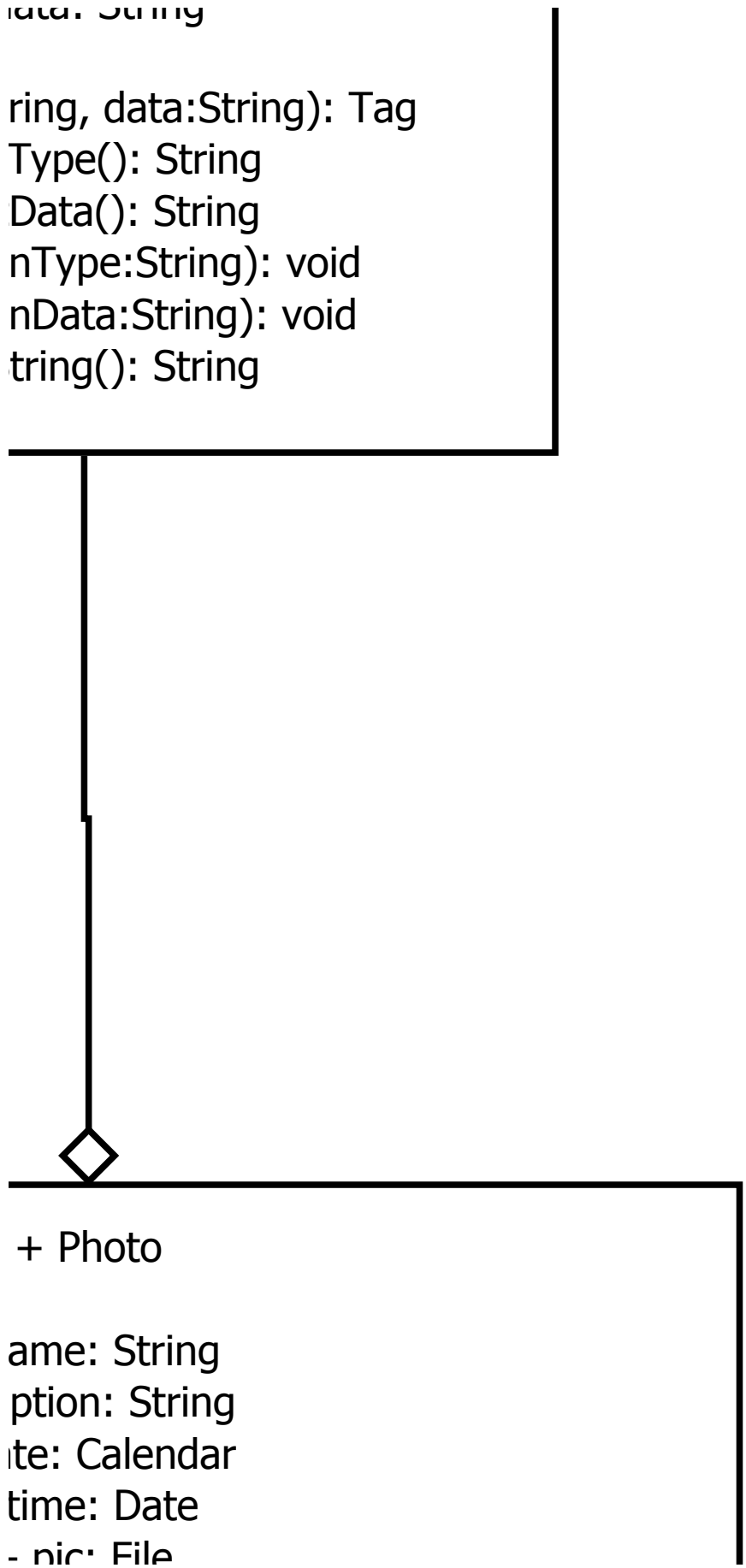

<<interface>>
+ BackendInterface
+ final storeDir: String
+ final storeFile: String
ıg, User> readUsers(): HashMap<String, User>
rs(userDB:HashMap<String, User>): void
deleteUser(userID:String): void
+ void addUser(u:User): void
Jser getUser(userID:String): User
tUserDB(): HashMap<String, User>
)B(userDB:HashMap<String, User>): void

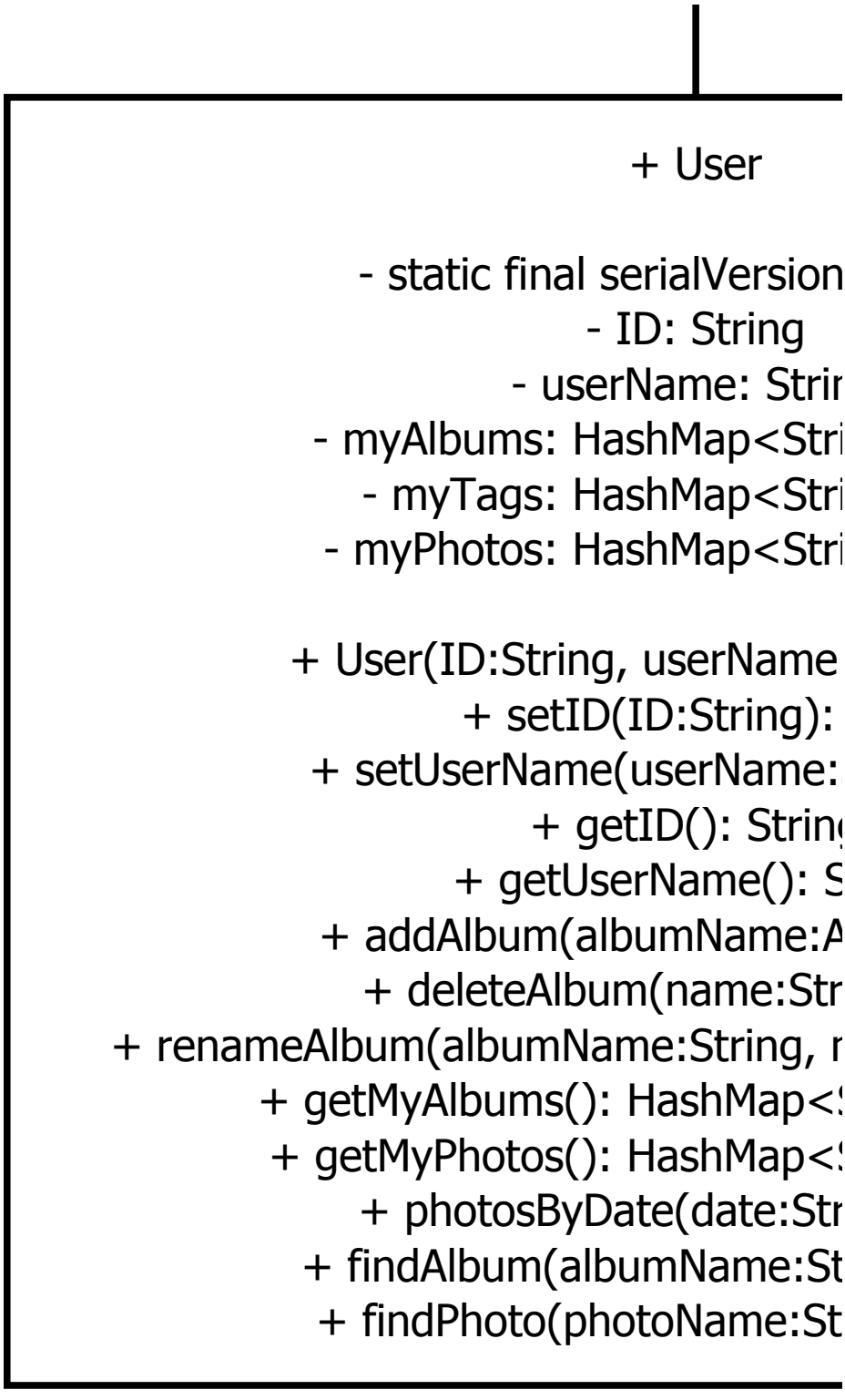

Backend

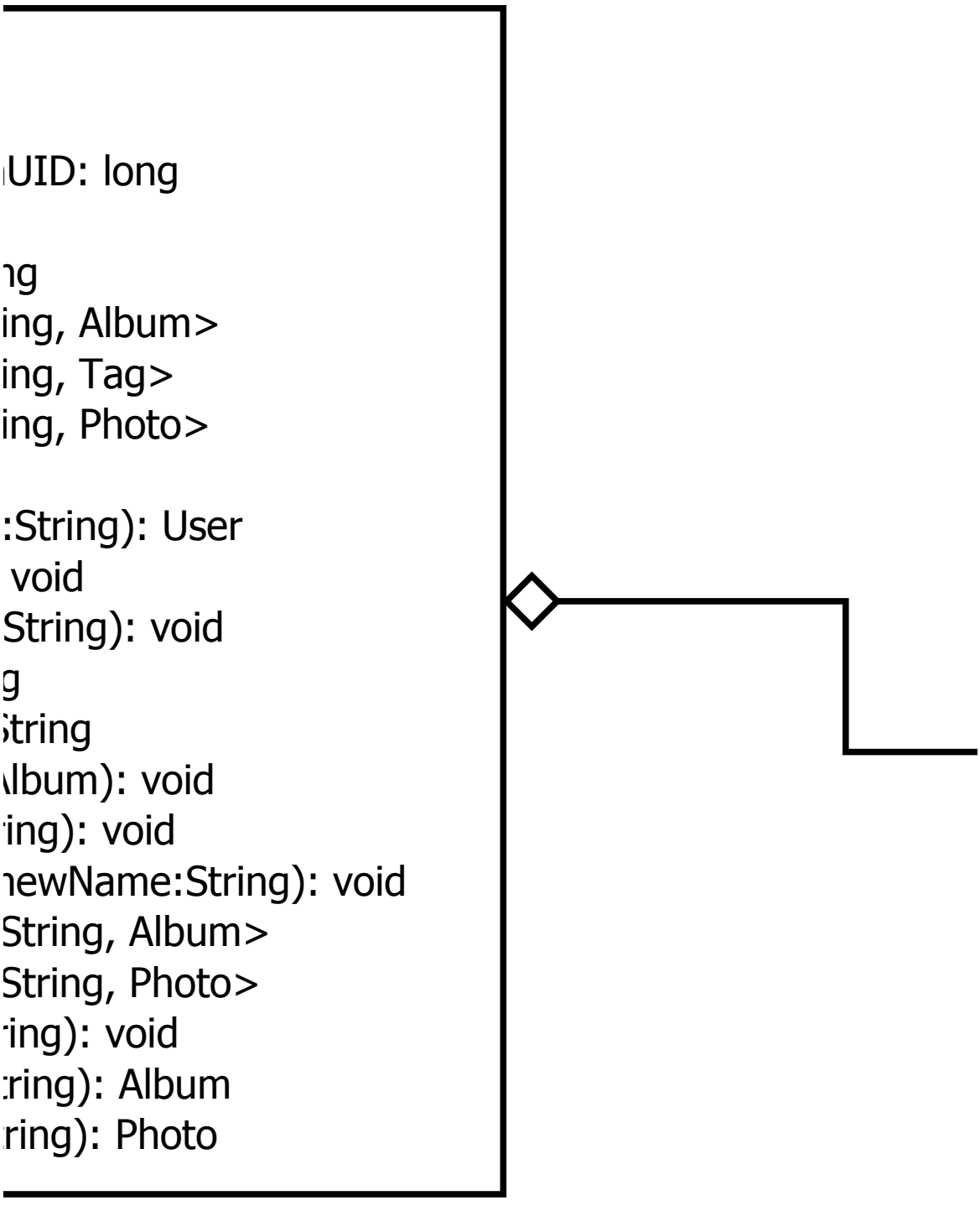
+ logout(): void
getStartDate(Album album): String
- getEndDate(Album album): String

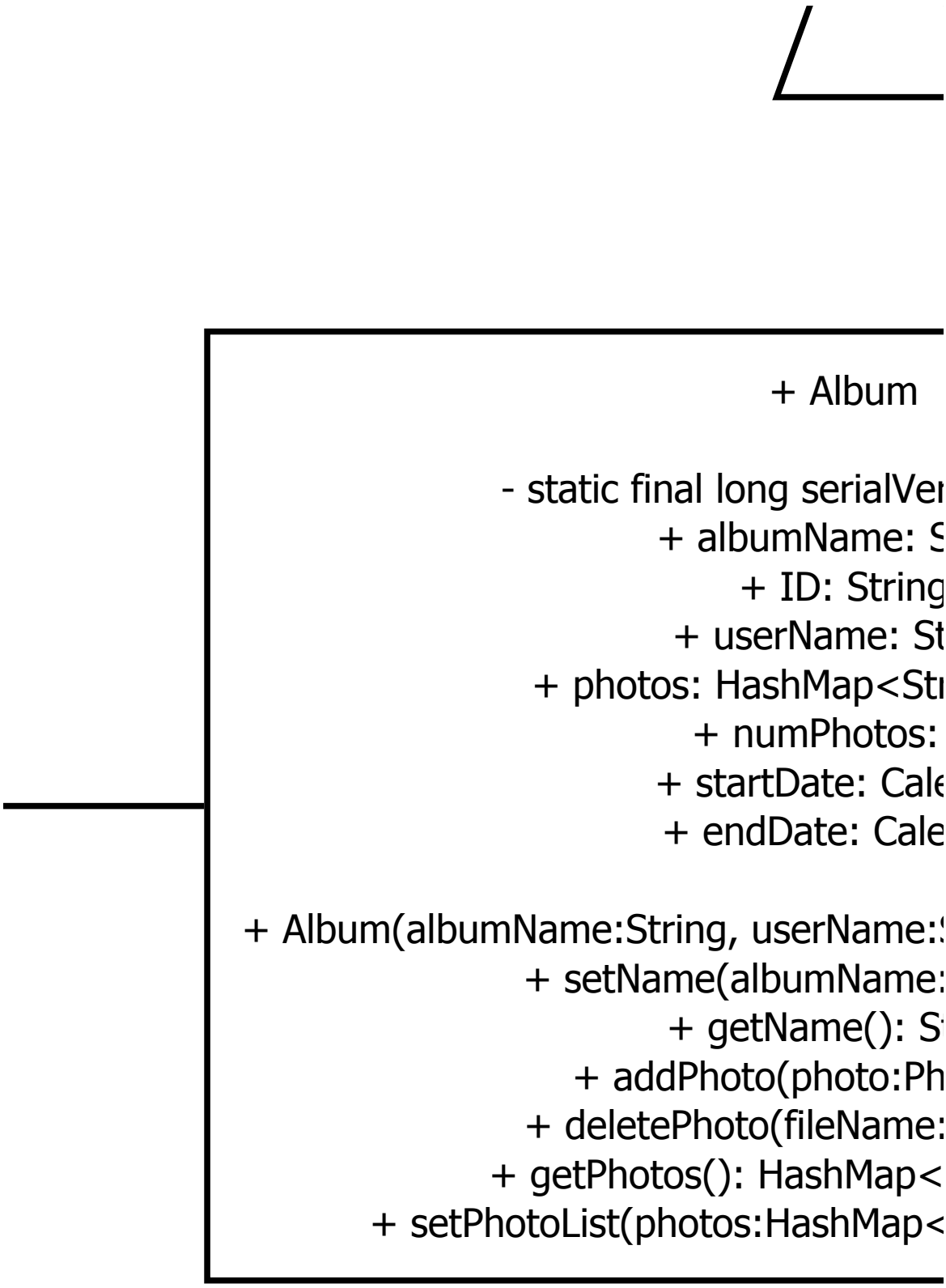


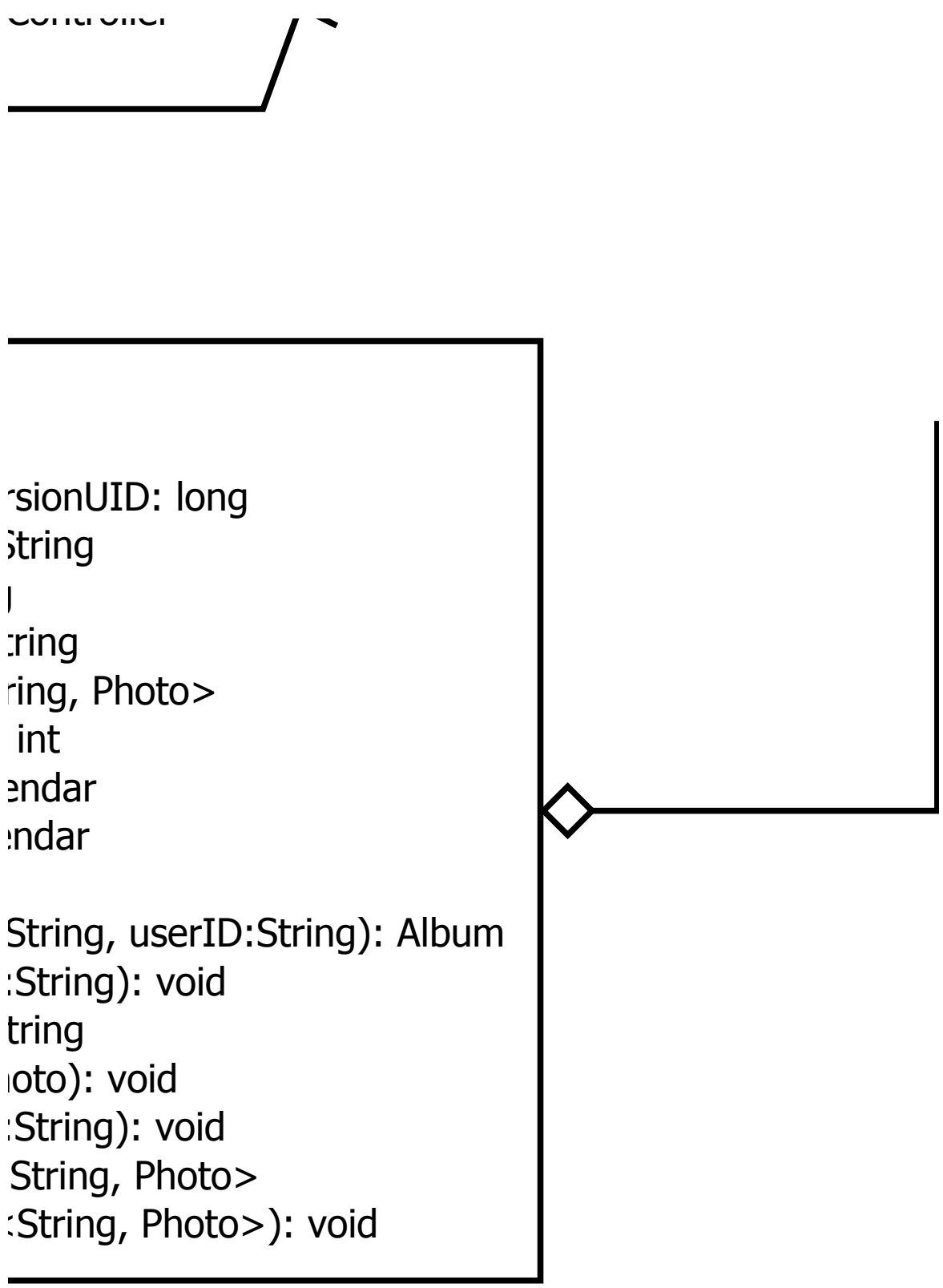

+ Tag(type:St
+ get
+ get
+ setType(
+ setData(
+ toS
mparator
gComparator
- n
- ca
- da

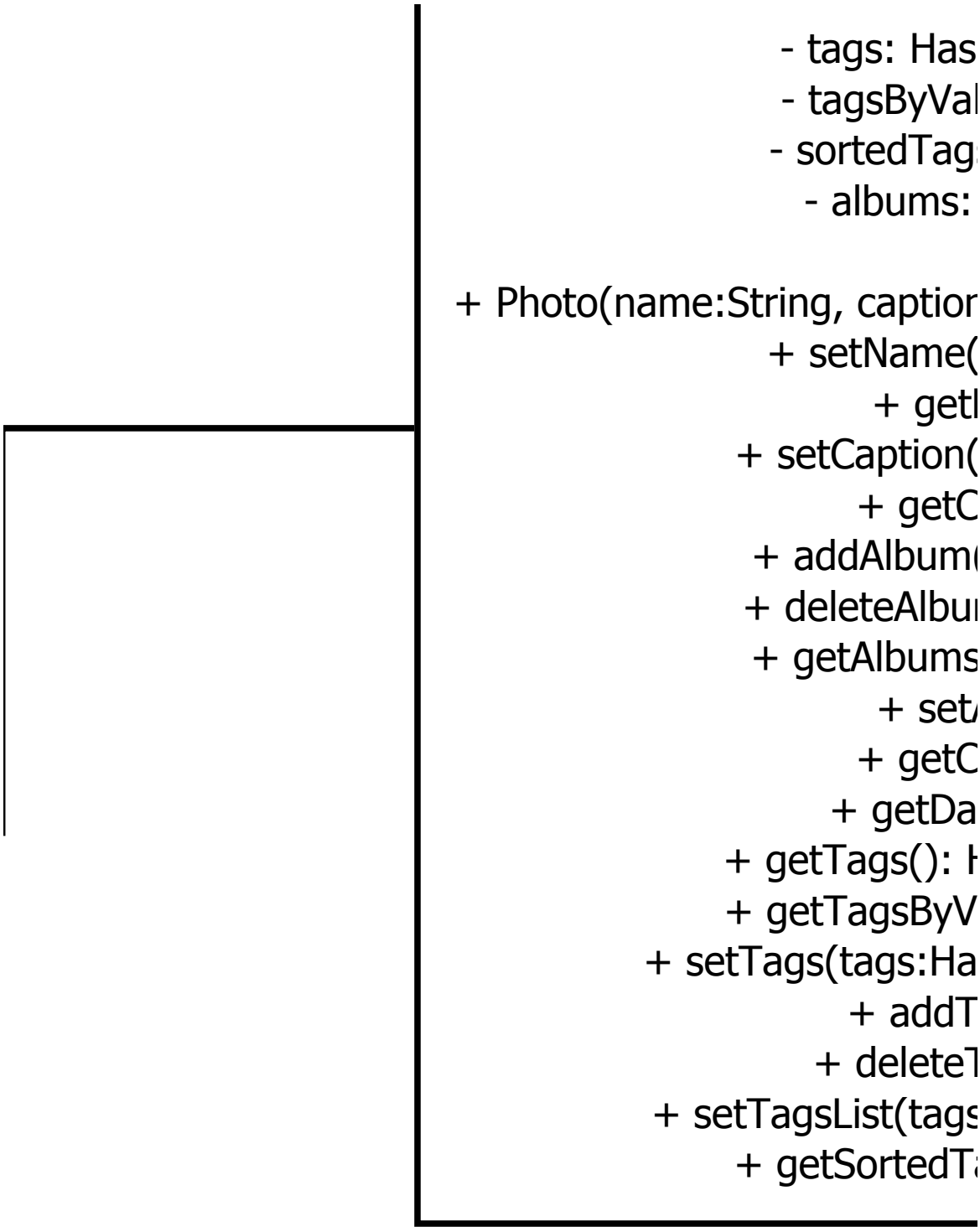

ring, data:String): Tag
Type(): String
Data(): String
nType:String): void
nData:String): void
tring(): String
+ Photo
ame: String
ption: String
te: Calendar
time: Date

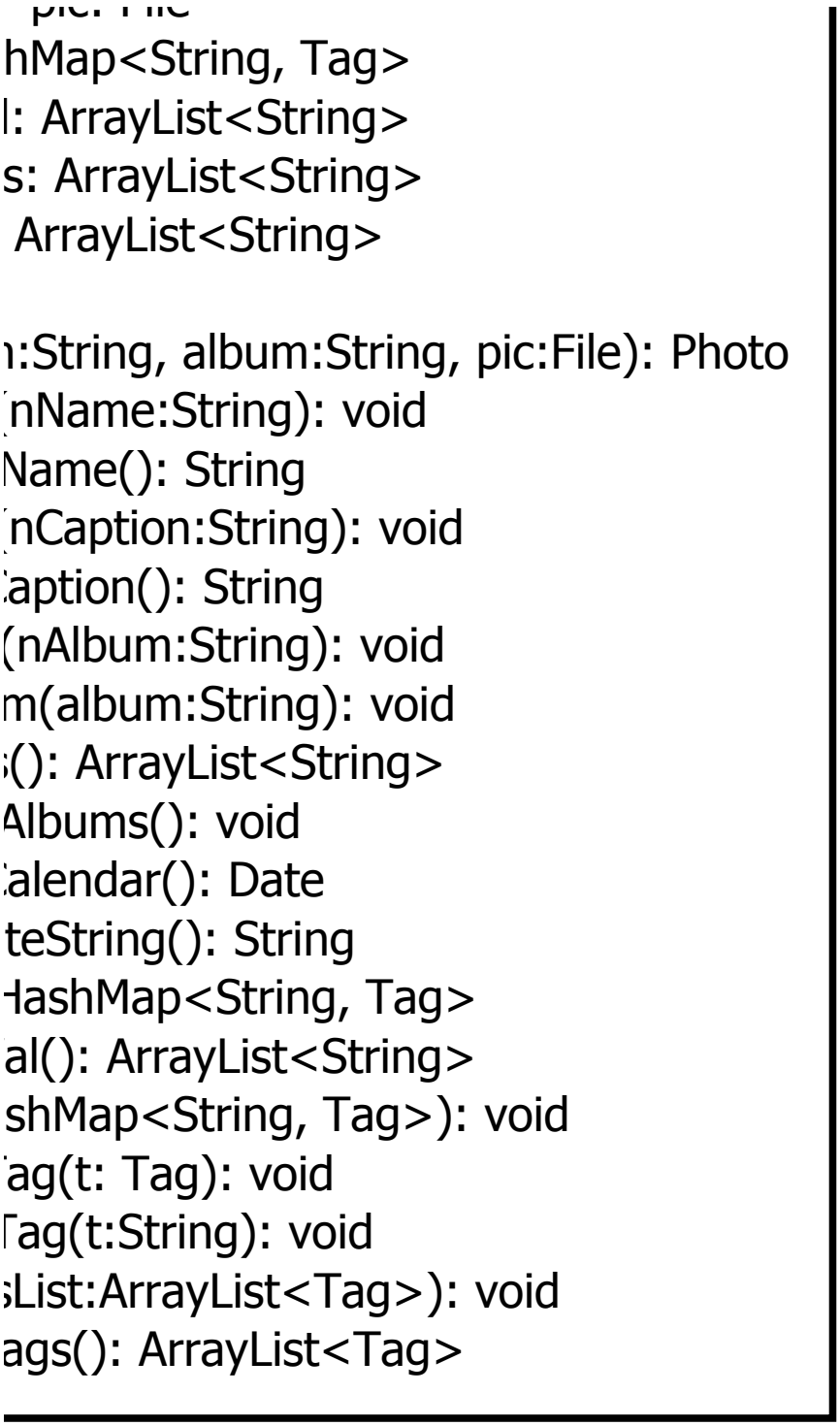

+ User
- static final serialVersion
- ID: String
- userName: Strir
- myAlbums: HashMap<Stri
- myTags: HashMap<Stri
- myPhotos: HashMap<Stri
+ User(ID:String, userName
+ setID(ID:String):
+ setUserName(userName:
+ getID(): String
+ getUserName(): S
+ addAlbum(albumName:A
+ deleteAlbum(name:Str
+ renameAlbum(albumName:String, r
+ getMyAlbums(): HashMap<S
+ getMyPhotos(): HashMap<S
+ photosByDate(date:Str
+ findAlbum(albumName:St
+ findPhoto(photoName:St


UID: long
ng
ing, Album>
ing, Tag>
ing, Photo>
:String): User
void
String): void
g
String
lbum): void
ing): void
newName:String): void
String, Album>
String, Photo>
ing): void
ring): Album
ring): Photo


+ Album
- static final long serialVer
+ albumName: S
+ ID: String
+ userName: St
+ photos: HashMap<Str
+ numPhotos:
+ startDate: Cale
+ endDate: Cale
+ Album(albumName:String, userName:S
+ setName(albumName:
+ getName(): St
+ addPhoto(photo:Ph
+ deletePhoto(fileName:
+ getPhotos(): HashMap<
+ setPhotoList(photos:HashMap<


rsionUID: long
String
ring
ring, Photo>
int
endar
endar
String, userID:String): Album
String): void
tring
oto): void
String): void
String, Photo>
String, Photo>): void


- tags: Has
- tagsByVal
- sortedTag
- albums:
+ Photo(name:String, captior
+ setName(
+ getl
+ setCaption(
+ getC
+ addAlbum(
+ deleteAlbu
+ getAlbums
+ set
+ getC
+ getDa
+ getTags(): H
+ getTagsByV
+ setTags(tags:Ha
+ addT
+ deleteT
+ setTagsList(tags
+ getSortedTa


pic: File
hMap<String, Tag>
l: ArrayList<String>
s: ArrayList<String>
ArrayList<String>
n:String, album:String, pic:File): Photo
nName:String): void
Name(): String
nCaption:String): void
aption(): String
(nAlbum:String): void
m(album:String): void
(): ArrayList<String>
Albums(): void
alendar(): Date
teString(): String
HashMap<String, Tag>
al(): ArrayList<String>
shMap<String, Tag>): void
ag(t: Tag): void
Tag(t:String): void
sList:ArrayList<Tag>): void
ags(): ArrayList<Tag>