## Post

## Story

- expiredTime: LocalDateTime
- available: boolean

## Content

#contentId:String

#String authorId;

#{readOnly}timestamp:LocalDateTime

# content : String

+ createContentId(): String
+ getAuthorId(): String
+ getContent(): String
+ getContentId(): String
+ getTimestamp(): LocalDateTime

## ContentDataBase

- contentDatabaseInstance: ContentDatabase

+ storeContent(imagePaths: String[], contentId: String): void
- getFileExtension(fileName: String): String
+ deleteContent(contentId: String): void
+ getInstance(): ContentDatabase

*retrieves*

## ImageResizer

+ resizeImage(imagePath: String, newWidth: int, newHeight: int): BufferedImage
+ loadImage(imagePath: String): BufferedImage

*Uses*

## NewsFeedPage

+ NewsfeedPage(contents: ArrayList<Content>)
+ addContent(content: Content): void
+ Fetch(): ArrayList<Content>
+ getUserContents(Id: String): ArrayList<Content>
+ getFriendsContents(id: String[]): ArrayList<Content>
+ getStories(): ArrayList<Content>
+ removeExpiredStories(): void
- isExpired(story: Story): boolean

*Fetches content for*

## BackEnd

- backEndInstance: BackEnd
- userDatabase: UserDatabase
- userCredentials: UserCredentials

+ checkEmail(email: String): Boolean
+ checkUsername(username: String): Boolean
+ createUser(email: String, password: String, userName: String, dateOfBirth: Date): void
+ validateUser(email: String, password: String): User
+ getUser(userId: String): User
+ createContent(userId: String, imagePaths: String[], contentType: String, description: String): void
+ getInstance(): BackEnd

*Interacts with*

## Profile

+ getBio(): String
+ getProfilePhoto(): String
+ getCoverPhoto(): String

+ setCoverPhoto(coverPhoto: String): void
+ setProfilePhoto(profilePhoto: String): void
+ setBio(bio: String): void

## User

+ getProfile(): Profile
+ addProfile(profile: Profile): void
+ getUserId(): String
+ getUsername(): String
+ getEmail(): String

*Manages*

## UserDatabase

+ storeUser(user: User): void
+ getUser(userId: String): User
+ getUserUsingEmail(email: String): User
+ findUserUsingUsername(userName: String): User
+ readJsonFile(): ArrayList<User>
+ writeJsonFile(users: ArrayList<User>): void
+ getInstance(): UserDatabase

*Validates credentials*

*retrieves*

*Validates*

## UserCredentials

+ addUserCredentials(email: String, password: String): void
+ checkUserCredentials(email: String, password: String): Boolean
+ findUser(email: String): Boolean
+ getUserCredentialsInstance(): UserCredentials

## «interface»
## Singelton

+ getInstance(): Singleton

*Uses*  *Uses*

## PasswordHashing

+ hashPassword(password: String): String
+ hashPassword(password: String, salt: String): String
- generateSalt(): byte[]

## EmailFormatChecker

+ isValidEmail(email: String): boolean

## FriendManagementDatabase

- friendDatabase: HashMap<String, FriendManagementDataBase>

- addFriendData(userId: String, friendData: FriendManagementDataBaze): void
- getFriendData(userId: String): FriendManagementDataBase
- updateFriendData(userId: String, friendData: FriendManagementDataBaze): void
- removeFriendData(userId: String): void

## FriendManagement

- backEnd: BackEnd
- friendManagementInstance: FriendManagement (Singleton)

- sendFriendRequest(senderName: String, receiverName: String): boolean
- acceptFriendRequest(senderName: String, receiverName: String): boolean
- declineFriendRequest(senderName: String, receiverName: String): boolean
- blockUser(senderName: String, receiverName: String): boolean
- unblockUser(senderName: String, receiverName: String): boolean
- getInstance(): FriendManagement (Singleton)

retrieves

## FriendManagementData

- userId: String
- friendsListIds: ArrayList<String>
- blockedListIds: ArrayList<String>
- requestListIds: ArrayList<String>
- pendingInvitesListIds: ArrayList<String>

- addToFriendList(friendId: String): void
- removeFromFriendList(friendId: String): void
- addToBlockedList(friendId: String): void
- removeFromBlockedList(friendId: String): void
- addToRequestList(friendId: String): void
- removeFromRequestList(friendId: String): void
- addToPendingInvitesList(friendId: String): void
- removeFromPendingInvitesList(friendId: String): void