MainUser mainUser : User

+ <u>setMainUser(</u>User): void + <u>getMainUser()</u>: User

ConnectHub Main(String): void

FriendRequest

requestSender: User - requestReceiver: User requestStatus: Status

- Accepted

+ Declined

<<enumeration>> Status

+ Pending + FriendRequest(User, User, Status): getRequestSender(): User getRequestReciever(): User getRequestStatus(): Status - setRequestSender(User): void setrequestReciever(User): void

setRequestStatus(Status): void

<u>HashingUtil</u> generateUserHashedPassword(String): string

FriendshipManagement

+ acceptFriendRequest(User,FriendRequest): boolean

+ declineFriendRequest(User,FriendRequest): boolean

+ sendFriendRequest(User,User): boolean

+ blockUser(User,User): void

+ removeFriend(User,User): void + suggestFriends(User): Set<User>

· validateUserInput(email: String, username: String, password: String, dateOfBirth: LocalDate): boolean validateUserEmail(email: String): boolean

> UserSignupSingleton - userInstance: UserSignupSingleton - user: User

UserSignupSingleton()

getUser(): User setUser(user: User): void getInstance(): UserSignupSingleton + userSignup(email: String, username: String, password: String, dateOfBirth: LocalDate): boolean

User

userId: String - email: String username: String password: String - dateOfBirth: String status: String friends: Set<User> - blockedUsers: Set<User> - sentFriendRequests: Set<FriendRequest> - receivedFriendRequests: Set<FriendRequest>

getUserId(): String + getUserEmail(): String + getUsername(): String getUserDateOfBirth(): String + getUserStatus(): String + getUserPassword(): String

+ userLogout(): void + toJSON(): JSONObject

+ blockUser(user: User): void + removeFriend(user: User): void + isUserBlocked(user: User): boolean

getUserFriends(): Set<User> + getUserBlockedUsers(): Set<User>
+ getUserSentFriendRequests(): Set<FriendRequest> + getUserReceivedFriendRequests(): Set<FriendRequest>
+ setUserPassword(unHashedPassword: String): void + setUserDateOfBirth(date: LocalDate): void + setUsername(username: String): void + setUserStatus(status: String): void + setUserEmail(email: String): boolean

+ acceptFriendRequest(request: FriendRequest): boolean + declineFriendRequest(request: FriendRequest): boolean

+ sendFriendRequest(receiver: User): void

UserDatabase

ValidationUtil

user_database: UserDatabase - users: ArrayList<User> - uniqueCounter: int · UserDatabase()

+ getUniqueCounter(): int + getInstance(): UserDatabase + getUserIds(): ArrayList<String> + getUsers(): ArrayList<User> + addUser(user: User): boolean + getUser(userId: String): User userLogin(userId: String, password: String): boolean

+ saveUsersToFile(filePath: String): void

+ getUserFromId(userId: String): User

+ loadUsersFromFile(filePath: String): void

ImageResizer - ImageResizer() + resizeImage(imagePath: String, width: int, height: int): ImageIcon

LoginFrame

+ LoginFrame() - initCustomComponents(): void + main(args: String[]): void
- passwordActionPerformed(evt: java.awt.event.ActionEvent): void

<<Swing Components>> - jLabel1: JLabel - jLabel2: JLabel - login: JButton - password: JTextField - userid: JTextField

FriendsGui

+ jLabel2MouseClicked(evt: java.awt.event.MouseEvent): void

+ uploadPostEvent(evt: java.awt.event.ActionEvent): void + uploadStoryEvent(evt: java.awt.event.ActionEvent): void

- currentUser: User - friendRequestsModel: DefaultListModel<String> - friendListModel: DefaultListModel<String> - friendSuggestionsModel: DefaultListModel<String> - changeFriendStatusModel: DefaultListModel<String>

selectedFile: File

+ CreateContentP() + initComponents(): void

<<Swing Components>> - jLabel2: JLabel

- jScrollPane1: JScrollPane - jTextArea1: JTextArea

- uploadPostButton: JButton

uploadStoryButton: JButton

+ FriendsGui() + initCustomComponents(): void - acceptActionPerformed(evt: java.awt.event.ActionEvent): void - declineActionPerformed(evt: java.awt.event.ActionEvent): void - addFriendActionPerformed(evt: java.awt.event.ActionEvent): void - blockActionPerformed(evt: java.awt.event.ActionEvent): void - removeActionPerformed(evt: java.awt.event.ActionEvent): void

<<Swing Components>> - accept: JButton - decline: JButton - addFriend: JButton - block: JButton - remove: JButton - friendRequests: JList<String> - friendList: JList<String> - friendSuggestions: JList<String> - changeFriendStatus: JList<String>
- jTabbedPane2: JTabbedPane - jPanel1: JPanel - jPanel2: JPanel - jPanel3: JPanel - jPanel4: JPanel - jScrollPane1: JScrollPane - jScrollPane2: JScrollPane - jScrollPane3: JScrollPane

- jScrollPane4: JScrollPane - jScrollPane5: JScrollPane

SignupFrame + SignupFrame() - initCustomComponents(): void - signupActionPerformed(evt: java.awt.event.ActionEvent): void + main(args: String[]): void

<<Swing Components>> email: JTextField - jCalendar1: JCalendar - jLabel1: JLabel - jLabel2: JLabel - jLabel3: JLabel - jLabel4: JLabel - password: JPasswordField

- signup: JButton

username: JTextField

WelcomeFrame

+ WelcomeFrame() - initCustomComponents(): void - loginActionPerformed(evt: java.awt.event.ActionEvent): void

+ main(args: String[]): void <<Swing Components>> - jLabel1: JLabel login: JButton

- signup: JButton

news - panel1: JPanel - panel2: JPanel - panel3: JPanel

panel4: JPanel

- panel5: JPane

+ news() throws IOException + postComp(post: Post): JPanel + sroryComp(story: Story): JPanel + friendComp(friend: User): JPanel + refreshContent(): void + createStyledButton(text: String): JButton

+ main(args: String[]): void <<Swing Components>> - button1: JButton - refreshButton: JButton - button3: JButton - tabbedPane: JTabbedPane

- buttonPanel: JPanel

AContent

- contentId: int - author: User text: String · imagePath: String - timeOfUpload: LocalDateTime - timeStampFormat: DateTimeFormatter

+ getTimestamp(): String

+ abstract uplode(): void

+ getText(): String + getImagePath(): String + getContentId(): int + getAuthor(): User + getTimeOfUpload(): LocalDateTime + setText(text: String): void

+ setImagePath(imagePath: String): void + setTimeOfUpload(timeOfUpload: LocalDateTime): void + setTimeOfUpload(): void + getTimeStampFormat(): DateTimeFormatter + AContent(author: User) + toJSON(): JSONObject

ContentDataBase

- posts: ArrayList<Post> stories: ArrayList<Story> - dataBase: ContentDataBase - id: int getInstance(): ContentDataBase addContent(cont: Post): void addContent(cont: Story): void removeStory(): void

getPosts(): ArrayList<Post> # getStories(): ArrayList<Story> getFriendsPosts(user: User): ArrayList<Post> + getFriendsStories(user: User): ArrayList<Story> getUniqueId(): int - load(): void + save(): void + close(): void

Post

+ Post(authorld: User) + uplode(): void + fromJSON(jsonObject: JSONObject): Post

+ Story(authorld: User) + uplode(): void + fromJSON(jsonObject: JSONObject): Story

story