Willkommen zu Locster: Soziales Netzwerk

Gliederung

- Allgemeines
- Idee
- Projektstruktur
- Klassenmodell
 - User
 - Guestbook
 - Chat
 - Friends
 - Utility
 - Exceptions
- Junit test
- Lessons Learned



Team: Dirk Hofmann, Jakob Gensel, Matthias Gabel, Molham Al-Khodari, Saskia Wohlers **GitHub Repository:** https://github.com/fh-erfurt/Java1-Locster



Idee

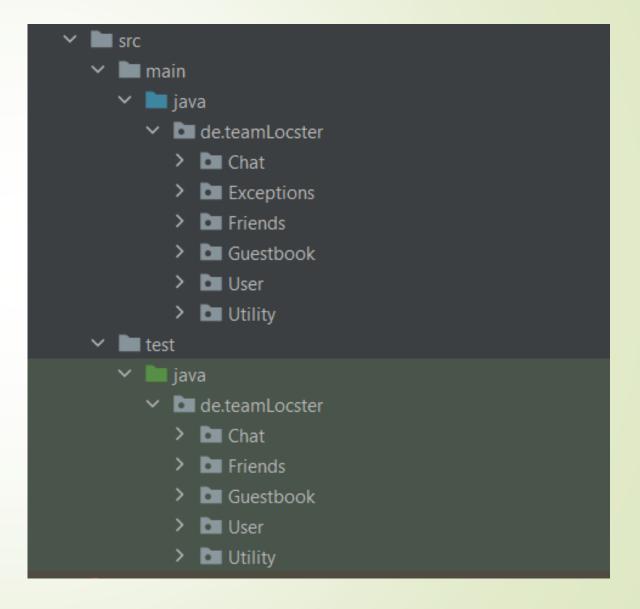
Soziales Netzwerk mit folgenden features:

- Chatfunktion
- Freundschaftsanfragen verschicken und akzeptieren
- User registrieren
- Profilstatistiken automatisch updaten
- Profiltext bearbeiten
- Gästebucheinträge schreiben

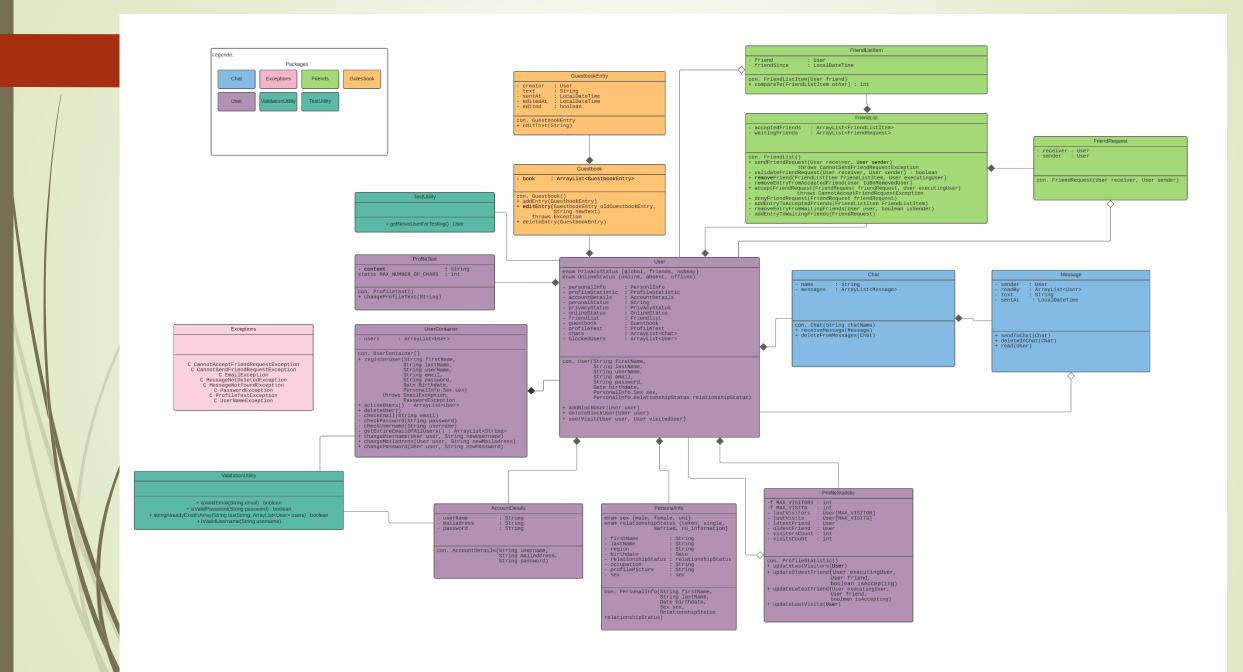


Projektstruktur





Klassenmodell



Løcster

Userverwaltung

```
ProfileText
                                                                                              User
- content
                           : String
                                                                 enum PrivacyStatus {global, friends, nobody}
static MAX_NUMBER_OF_CHARS : int
                                                                 enum OnlineStatus {online, absent, offline}
                                                                 - personalInfo
                                                                                    : PersonlInfo
con. Profiletext()
                                                                 - profileStatistic : ProfileStatistic
+ changeProfileText(String)
                                                                 - accountDetails
                                                                                    : AccountDetails
                                                                 - peronalStatus
                                                                                    : String
                                                                 - privacyStatus
                                                                                    : PrivacyStatus
                                                                 - onlineStatus
                                                                                    : OnlineStatus
                                                                 - friendlist
                                                                                    : Friendlist
                                                                 - questbook
                                                                                    : Guestbook
                     UserContainer
                                                                 - profileText
                                                                                    : ProfileText
                                                                 - chats
                                                                                    : ArrayList<Chat>
            : ArrayList<User>
                                                                 - blockedUsers
                                                                                    : ArrayList<User>
- users
con. UserContainer()
+ registerUser(String firstName,
                                                                 con. User(String firstName,
              String lastName,
                                                                          String lastName,
              String userName,
                                                                          String userName,
              String email,
                                                                          String email,
              String password,
                                                                          String password,
              Date Birthdate,
                                                                          Date birthdate,
              PersonalInfo.Sex sex)
                                                                          PersonalInfo.Sex sex,
       throws EmailException,
                                                                          PersonalInfo.RelationshipStatus relationshipStatus)
              PasswordException
+ activeUsers() : ArrayList<User>
                                                                + addBlockUser(User user)
+ deleteUser()
                                                                 + deleteBlockUser(User user)
- checkEmail(String email)
                                                                 + userVisit(User user, User visitedUser)
- checkPassword(String password)
- checkUsername(String username)
- getEntireEmailOfAllŬsers() : ÁrrayList<String>
+ changeUsername(User user, String newUsername)
+ changeMailadress(User user, String newMailadress)
+ changePassword(User user, String newPassword)
```

AccountDetails

- userName : String - mailadress : String password : String

con. AccountDetails(String username, String mailAddress, String password)



PersonalInfo

enum sex {male, female, uni} enum relationshipStatus {taken, single, married, no information}

: String - firstName - lastName : String - region : String - birthdate : Date

- relationshipStatus : relationshipStatus

- occupation : String - profilePicture : String - sex : sex

con. PersonalInfo(String firstName, String lastName, Date birthdate, Sex sex, RelationshipStatus

relationshipStatus)

ProfileStatistic

-f MAX_VISITORS : int

-f MAX VISITS : int

- lastVisitors : User[MAX_VISITOR] - lastVisits : User[MAX_VISITS]

- latestFriend : User

- oldestFriend : User

- visitorsCount : int

- visitsCount : int

con. ProfileStatistic()

+ updateLastVisitors(User)

+ updateOldestFriend(User executingUser,

User friend,

boolean isAccepting)

+ updateLatestFriend(User executingUser,

User friend,

boolean isAccepting)

+ updateLastVisits(User)

Løcster

Friendsverwaltung



- friend : User

- friendSince : LocalDateTime

con. FriendListItem(User friend)

+ compare(FriendListItem other): int



FriendList

- acceptedFriends : ArrayList<FriendListItem>

- waitingFriends : ArrayList<FriendRequest>

con. FriendList()

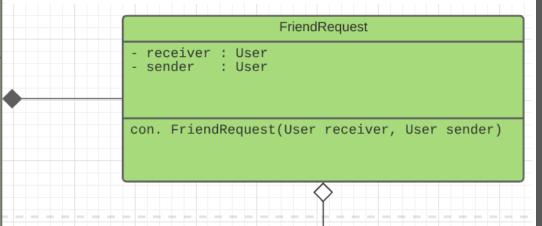
+ sendFriendRequest(User receiver, User sender)

throws CannotSendFriendRequestException

- validateFriendRequest(User receiver, User sender) : boolean
- + removeFriend(FriendListItem friendListItem, User executingUser)
- removeEntryFromAcceptedFriend(User toBeRemovedUser)
- + acceptFriendRequest(FriendRequest friendRequest, User executingUser)
 throws CannotAcceptFriendRequestException
- + denyFriendRequest(FriendRequest friendRequest)
- addEntryToAcceptedFriends(FriendListItem FriendListItem)
- removeEntryFromWaitingFriènds(User user, boolean isSendér)
- addEntryToWaitingFriends(FriendRequest)

FriendList ist der Mittelpunkt mit allen Funktionalitäten.

- in acceptedFriends befinden sich bestätigte Freunde
- in waitingFriends befinden sich Freundschaftsanfragen Funktionalitäten:
- Freundschaftsanfragen schicken
- Freundschaftsanfragen akzeptieren/verweigern
- Freund entfernen
- Einschränkungen/Ausnahmen wurden bedacht



Guestbookverwaltung





GuestbookEntry

- creator : User

- text : String
- sentAt : LocalDateTime
- editedAt : LocalDateTime

- edited : boolean

con. GuestbookEntry + editText(String)

Guestbook

- book : ArrayList<GuestbookEntry>

con. Guestbook()

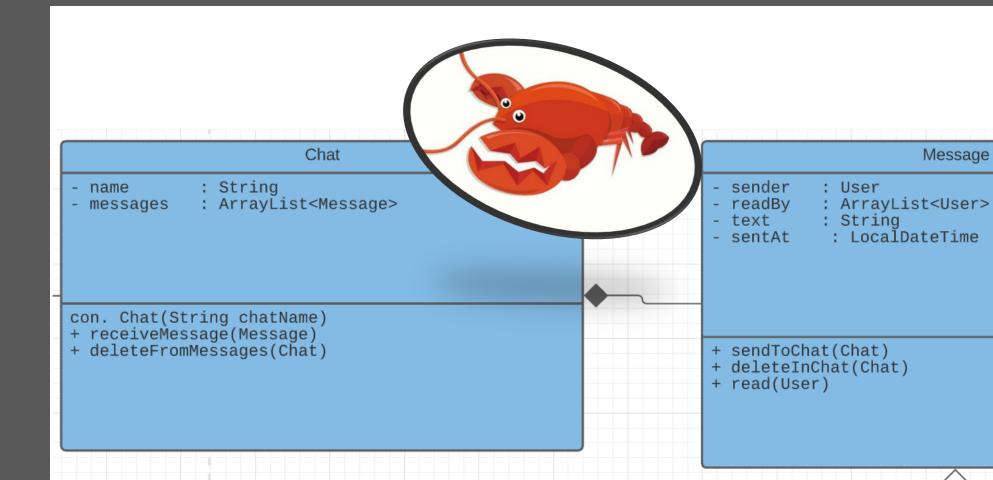
+ addEntry(GuestbookEntry)
+ editEntry(GuestbookEntry oldGuestbookEntry, String newText)

throws Exception

+ deleteEntry(GuestbookEntry)

Løcster

Chatverwaltung



Utility & Exceptions

Løcster

Exceptions

C CannotAcceptFriendRequestException
C CannotSendFriendRequestException
C EmailException
C MessageNotDeletedException
C MessageNotFoundException
C PasswordException
C ProfileTextException
C UserNameException

ValidationUtility

+ isValidEmail(String email) : boolean + isValidPassword(String password) : boolean + stringAlreadyExistInArray(String testString, ArrayList<User> users) : boolean + isValidUsername(String username)

TestUtility

+ getNewsUserForTesting(): User



Tests passed: 50 of 50 tests – 160 ms.

- ProfileStatisticTest
 - ✓ should_increase_totals_value_by_one()
 - ✓ should_return_new_visitors()
 - ✓ should_return_new_Visits()
- ✓ ✓ ChatTest

 - chat_should_receive_message()
 - ✓ should_create_a_chat_room()



- ValidationUtilityTest
 - should_test_password()

✓ should_change_mailadress_to_bjoernathotmailde()

✓ should_change_password_to_wurstwasser()

✓ should_change_username_to_mickey()

- should_check_if_email_is_already_in_use()
- should_test_email()
- PersonalInfoTest
 - should_change_region()
 - ✓ should_change_profilePicture()
 - should_change_occupation()
 - should_change_relationshipStatus()

- ✓ when_deleted_entry_should_not_be_found_anymore()
- ✓ when_edited_entry_should_have_new_text_flag_and_time()
- ✓ added_entry_should_appear_in_guestbook()

- ✓ MessageTest
 - ✓ message_should_contain_given_text()
 - message_should_be_saved_to_and_deleted_from_given_chat()
 - comparing_datetime_should_return_relative_value()

- ✓ UserContainerTest
 - ✓ should_change_password_from_user()
 - ✓ should_throw_PasswordException()
 - ✓ force_email_already_exist_exception()

 - √ force_email_invalid_exception()
 - ✓ should_change_username_from_user()
- ✓ should_delete_user()
- ✓ activeUser should contain all active user() ✓ should_test_if_email_is_correct_and_unique()
- ✓ should test_if_password_is_valid()
- ✓ should_change_email_from_user()
- ✓ should_store_all_emails_in_one_list()
- ✓ should_register_max_mustermann()
- ✓ FriendListTest
 - should_accept_friend_request_with_more_users()
 - should_deny_friend_request_from_userA() should_deny_friend_request_from_userB()
 - should_remove_accepted_friend_from_userA()
 - should_remove_accepted_friend_from_userB()
 - should_throw_CannotAcceptFriendRequestException()
 - ✓ should_accept_friend_request()
 - should_send_friend_requests_with_more_users()
 - ✓ should_return_correct_sender_and_receiver() should_remove_accepted_friend_with_more_users()
 - should_send_friend_request()

 - should_throw_CannotSendFriendRequestException()
- UserTest
 - should_add_User_to_blocked_Users()
 - should_print_a_user()
 - should_delete_user_out_of_blocked_users()
 - should change privacy status()



Testen ist geil



Projektstrukturierung



Feste Aufgabenverteilung



Objektstruktur ohne Anwendung



Umgang mit Github

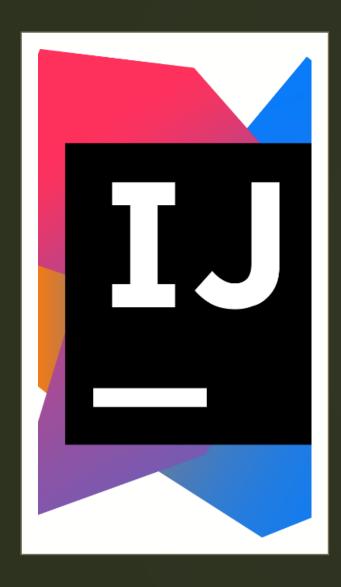


Kleinigkeiten sind keine Kleinigkeiten



Grundlagen Java

Lessons Learned









Tools

- IntelliJ IDEA
- Maven
- Junit 5
- Lucidchart
- Miro
- Discord
- GitHub

Danke für ihre Aufmerksamkeit









