



Willkommen zu Locster : Soziales Netzwerk

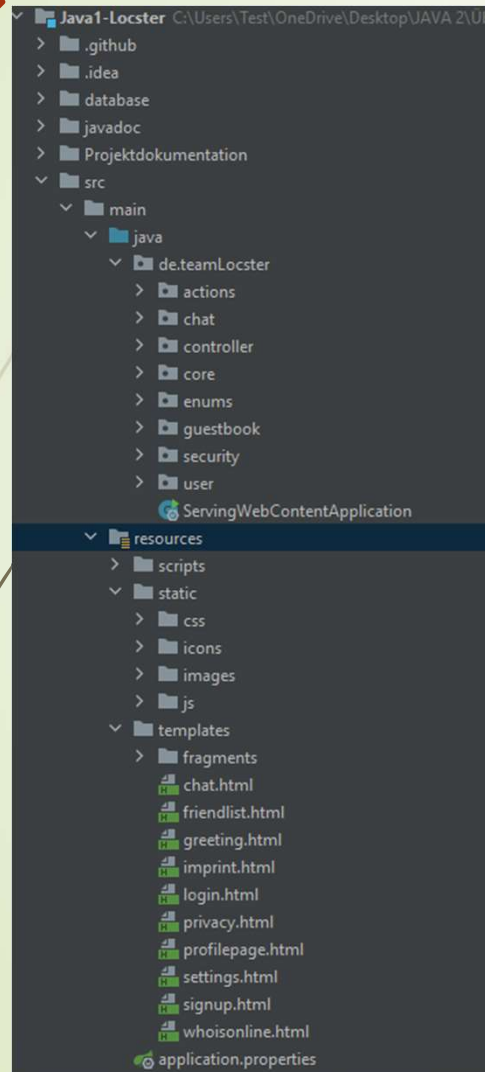


Idee

Soziales Netzwerk mit folgenden features:

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

Projektstruktur



/java: backend

/actions: Funktionen mit den User miteinander interagieren können. Z.B. sendFriendRequest()

/chat: Funktionen die den Chat betreffen
z.B. createChat()

/controller: Schnittstellen die Nutzeranfragen entgegennehmen
z.B. getLoginForm()

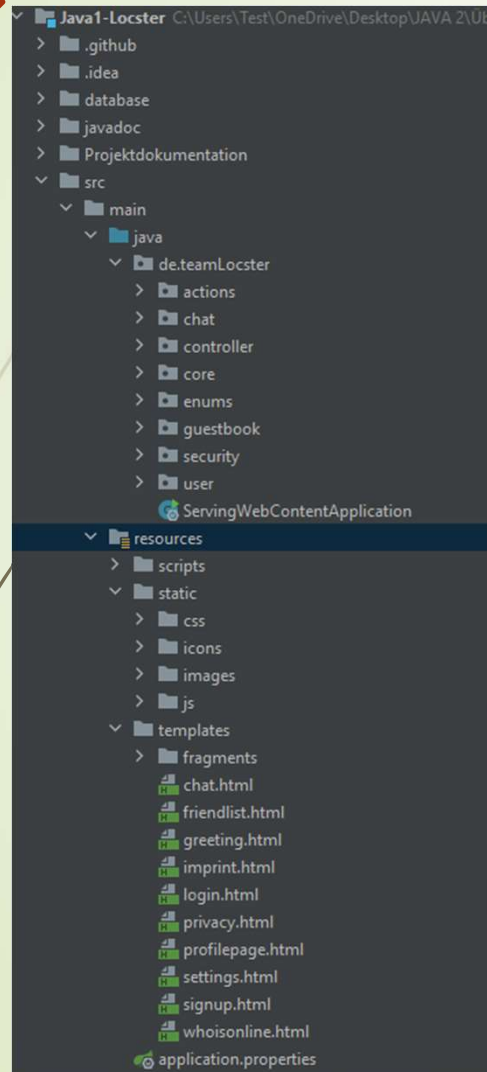
/enums: Enumerationen für die verschiedenen anwendungsfälle
z.B. biologisches Geschlecht (male, female, other)

/guestbook: spezifische Funktionen für das User Gästebuch

/security: Zugriffskonfiguration z.B. Login-funktionalität

/user: spezifische Funktionen für User z.B. Registrierung und DTO

Projektstruktur



/resources: frontend

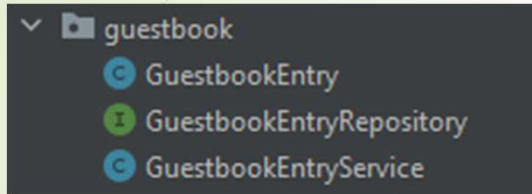
/scripts: framework Ordner

/static: styling-dateien, bilder/icons und scripte

/templates: html-dateien

application.properties = Konfigurationsdatei für DB's

Projektstruktur

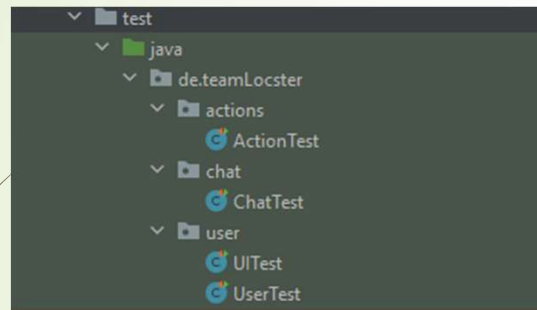


GuestbookEntry = model

GuestbookEntryRepository = schnittstelle (DB abfrage)

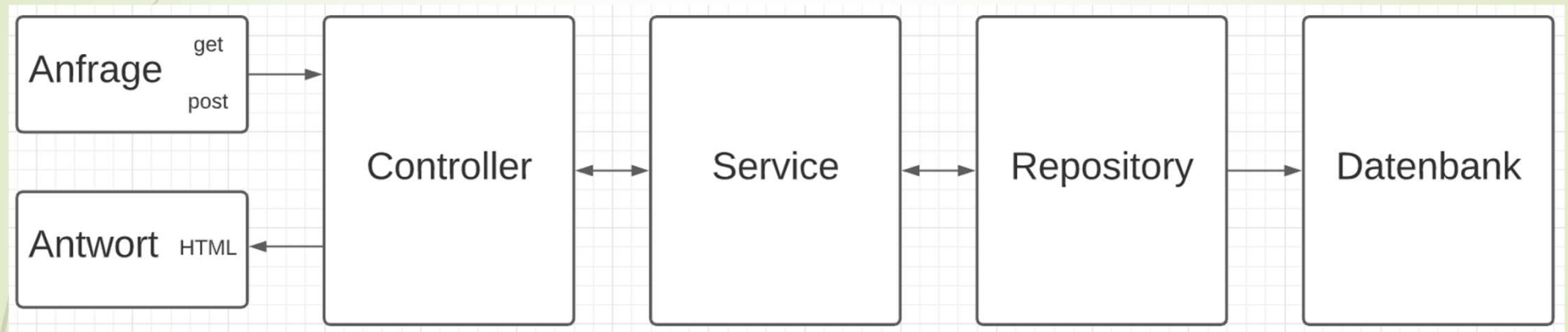
GuestbookEntryService = funktionalität

Projektstruktur



/test: verschiedene tests

Softwarearchitektur

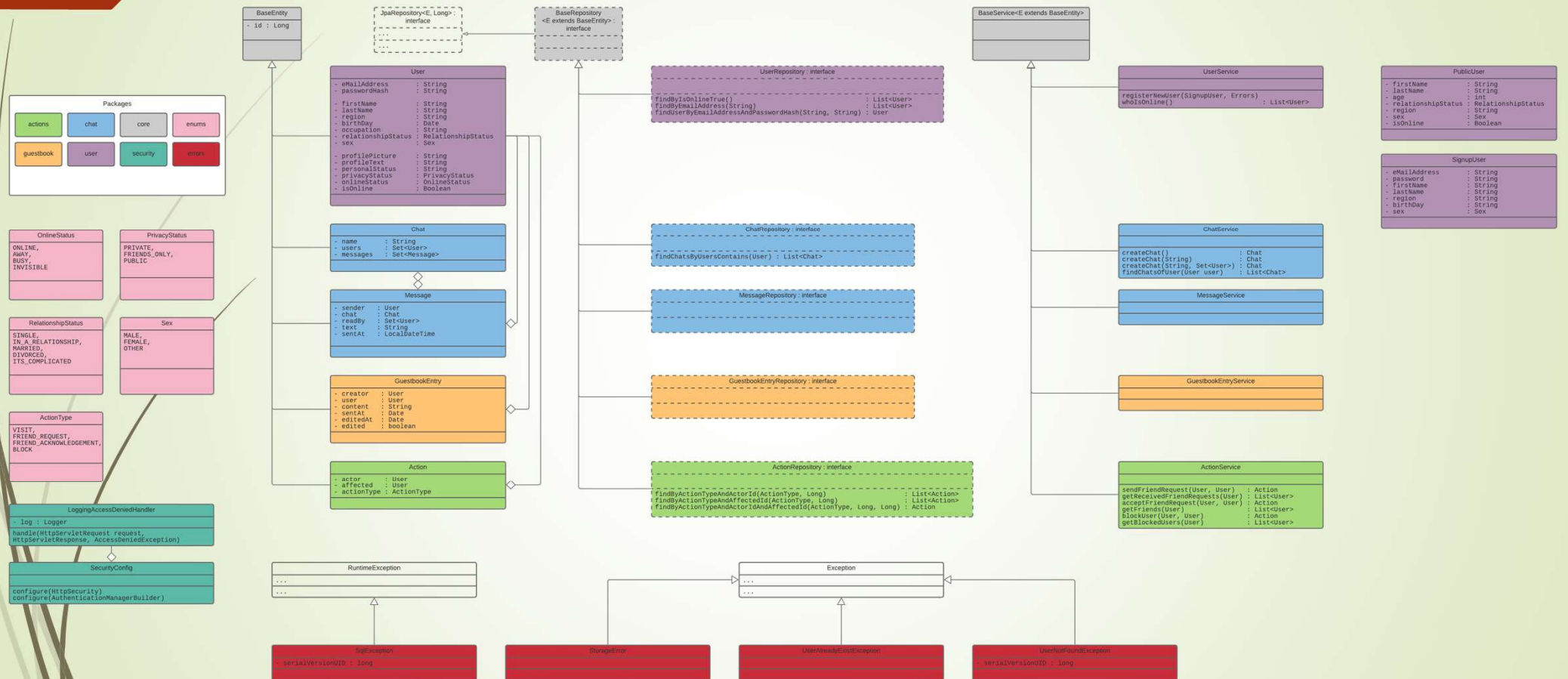


Komponenten

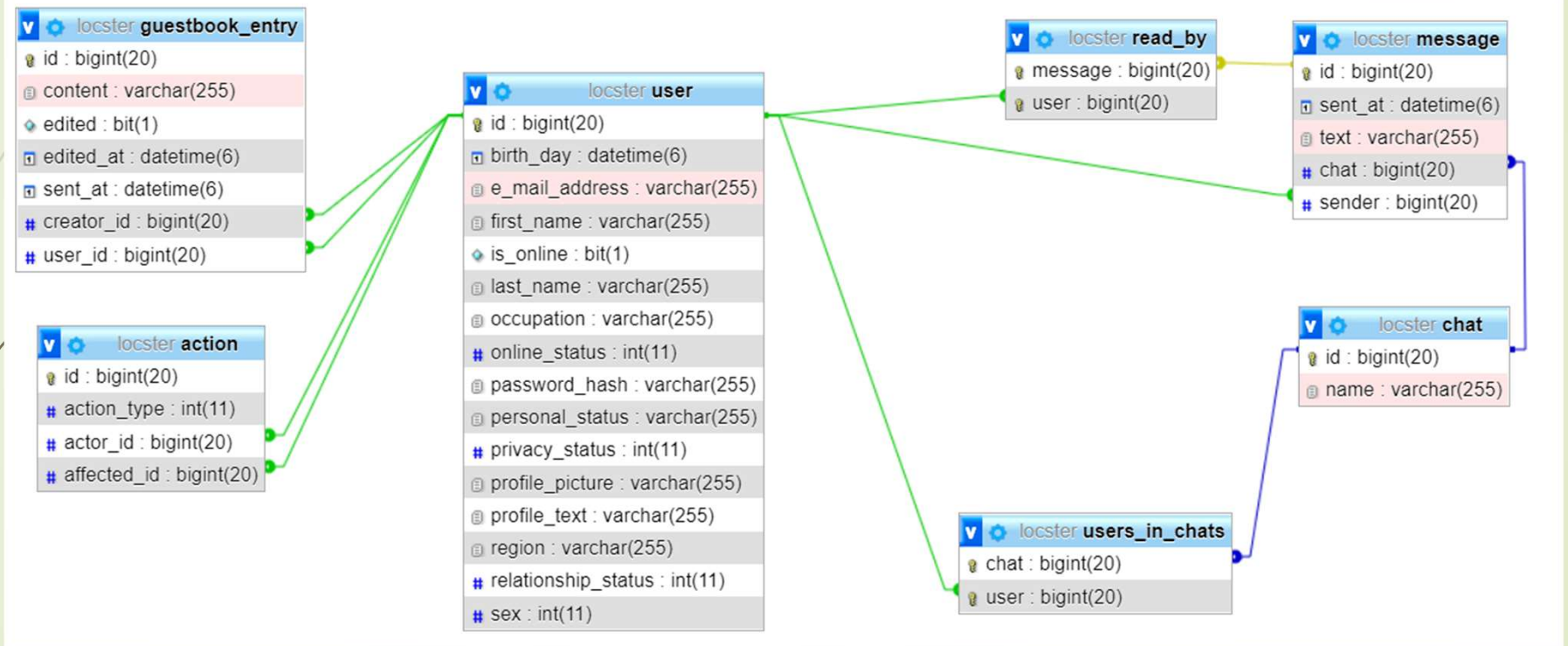


- DB-System: MariaDB
- Backend: JPA/Hibernate mit REST API, Spring
- Frontend: HTML/CSS/JS mit Thymeleaf

Klassendiagramm



DB Schema



Code Beispiel

```
35 @ public Action acceptFriendRequest(User actor, User affected) {  
36     Action result = null;  
37     Action friendRequest = actionRepository.findByActionTypeAndActorIdAndAffectedId(ActionType.FRIEND_REQUEST, affected.getId(), actor.getId());  
38     if (friendRequest != null) {  
39         result = actionRepository.save(new Action(actor, affected, ActionType.FRIEND_ACKNOWLEDGEMENT));  
40         // actionRepository.delete(friendRequest); // TODO sollen Anfragen nach Bestätigung gelöscht werden?  
41     }  
42     return result;  
43 }
```

```
1 <!DOCTYPE html>  
2 <html lang="en" xmlns:th="http://www.thymeleaf.org">  
3     <head th:replace="fragments/head"> </head>  
4  
5     <head>  
6         <link th:href = "{/css/login.css}" rel="stylesheet">  
7         <link th:href = "{/css/signup.css}" rel="stylesheet">  
8         <script type="text/javascript" th:src = "{/js/signup.js}"></script>  
9         <title>Locster.de.SignUp</title>  
10    </head>
```

Code Beispiel

```
19 function onPasswordChanged() {
20     const password = inputPassword.value;
21     if(password) {
22         let hasLength      = password.length >= 8;
23         let hasLowerCase    = password.match(/[a-z]/m);
24         let hasUpperCase    = password.match(/[A-Z]/m);
25         let hasNumber       = password.match(/[0-9]/m);
26         let hasSpecial      = password.match(/[@#^!"'$$%&/()=*\-+,.;:_<>?|\\{}\[\\\]]/m);
27         if (hasLength && hasLowerCase && hasUpperCase && hasNumber && hasSpecial) {
28             inputPassword.style.border = '2px solid green';
29         } else if (hasLowerCase && hasUpperCase && hasNumber && hasSpecial) {
30             inputPassword.style.border = '2px solid yellow';
31         } else if (hasLowerCase && hasUpperCase) {
32             inputPassword.style.border = '2px solid orange';
33         } else {
34             inputPassword.style.border = '2px solid red';
35         }
36         hasLength      ? length.style.color = 'green' : length.style.color = 'red';
37         hasLowerCase    ? minor.style.color = 'green' : minor.style.color = 'red';
38         hasUpperCase    ? major.style.color = 'green' : major.style.color = 'red';
39         hasNumber       ? number.style.color = 'green' : number.style.color = 'red';
40         hasSpecial      ? special.style.color = 'green' : special.style.color = 'red';
41     }
42 }
```

Webseiten



Live Demo

todo:



➤ Funktionen

➤ User

- Login/logout
- Daten vom Profil anzeigen/ändern
- Kontoeinstellungen
- Nutzer Besuchen
- Freundschaftsanfragen senden/erhalten/bestätigen
- Gästebucheinträge schreiben/bearbeiten/löschen
- Wer ist Online
- Suchfunktion
- Profiltext anzeigen/erstellen/ändern/löschen

➤ Messenger (optional)

- Chat-Nachrichten anzeigen/schreiben/bearbeiten/löschen
- Chat erstellen/löschen

Lessons learned



- Realistische Ziele setzen
- Erst nachdenken, dann Fehler machen
- Modularer Aufbau ist wichtiger als viele Features zuerst
- Der Nutzen von @Autowired und Annotations generell
- Spring (Boot) und Thymeleaf
- Zusammenspiel REST, Serverlogik & DB

Danke für ihre Aufmerksamkeit

