

Приложение WReview

ИЛИ
ЛУЧШИЙ РАБОТНИК ГОДА



Автор: Демьяненко Дмитрий Сергеевич, инженер-электроник КБФГ

Проблема оценки труда

Вот и настал конец года – время поощрения трудящихся. Но кто же все-таки удостоен быть награжденным?

- Как понять кто работал усерднее всех?
- Кому полагается дополнительное поощрения за свои труды?
- Чью фотографию повесить на доску почета?
- Чьего мнения придерживаться в решении этого вопроса?

Ответ прост: приложение **WReview**

3 этапа использования приложения

✓ 1 - Регистрация

The screenshot shows the registration interface of the 'Отзовик' (Otzovik) application. At the top, the logo 'Отзовик' is displayed with three colored icons (red sad face, yellow neutral face, green happy face) below it. Below the logo are three buttons: 'Войти' (Login), 'Регистрация' (Registration), and 'Рейтинг рабочих' (Workers' Rating). The 'Регистрация' button is highlighted. The registration form itself is titled 'Регистрация' and contains several input fields: 'Мои Ф.И.О.' (My F.I.O.) with the value 'Лампочкин Анатолий Валерьевич', 'Мой пароль' (My password) with a masked input, 'Моя дата рождения' (My date of birth) with the value '12.11.1994' and a calendar icon, 'Мой телефон' (My phone) with the value '89990123456', a checkbox for 'Я предоставляю услуги' (I provide services) which is checked, 'Описание услуг' (Description of services) with the text 'Электромонтёр, лампочку поменяю на раз-два' (Electrician, I will change the light bulb in one or two), 'Дата начала трудового стажа' (Date of start of work experience) with the value '05.06.2010' and a calendar icon, and 'Стоимость услуги' (Service cost) with the value '1500'. At the bottom of the form is a yellow button labeled 'Зарегистрироваться' (Register).

Отзовик

Войти Регистрация

Рейтинг рабочих

Регистрация

Мои Ф.И.О.: Лампочкин Анатолий Валерьевич

Мой пароль: *

Моя дата рождения: 12.11.1994

Мой телефон: 89990123456

Я предоставляю услуги: ☒

Описание услуг: Электромонтёр, лампочку поменяю на раз-два

Дата начала трудового стажа: 05.06.2010

Стоимость услуги: 1500

Зарегистрироваться

3 этапа использования приложения

✓ 2 – Поиск работника и его вызов

Отзовик

Вы вошли как: Лампочкин Анатолий Валерьевич

Выйти

Мои настройки

Рейтинг рабочих

Список рабочих

Все отзывы

Мои отзывы

Отзывы на меня

Добавить отзыв

Список рабочих

слесарь

Поиск

Очистить фильтр

Активен фильтр

Имя работника	Возраст	Услуги	Телефон	Стаж	Цена услуги	Рейтинг		
Лентяйкин Владимир Александрович	33	Слесарь-механик	89990000001	16	0	7,0	Подробнее	Написать отзыв
Трудоголиков Иван Иванович	43	Слесарь-сантехник	89990000000	25	0	2,0	Подробнее	Написать отзыв

3 этапа использования приложения

✓ 3 – Написать отзыв

Отзовик

Вы вошли как: Лампочкин Анатолий Валерьевич

Выйти

Мои настройки

Рейтинг рабочих

Список рабочих

Все отзывы

Мои отзывы

Отзывы на меня

Добавить отзыв

Добавить отзыв

Заказчик:

Лампочкин Анатолий Валерьевич

Дата оформления отзыва:

04.12.2023 10:39

Название оборудования:

Душевая цеха

Причина вызова:

Плохой напор воды, невозможно помыться после смены

Имя рабочего:

Трудоголиков Иван Иванович

Работа выполнена:

☒

Комментарий:

Работу выполнил качественно, спасибо от всего цеха!

Оценка рабочего:

10

Добавить отзыв


Приложение автоматически составит рейтинг рабочих за год

Вы вошли как: Лампочкин Анатолий Валерьевич [Выйти](#) [Мои настройки](#)

[Рейтинг рабочих](#) [Список рабочих](#) [Все отзывы](#) [Мои отзывы](#) [Отзывы на меня](#) [Добавить отзыв](#)

Рейтинг рабочих

Место	Имя работника	Рейтинг за год	Число отзывов за год	
1	Лопатин Антон Николаевич	10,0	1	Подробнее
2	Работина Юлия Сергеевна	9,0	1	Подробнее
3	Тарелкина Анастасия Борисовна	8,0	1	Подробнее
4	Лентяйкин Владимир Александрович	7,0	1	Подробнее
5	Трудоголиков Иван Иванович	6,0	2	Подробнее
6	Беспалый Юрий Дмитриевич	6,0	1	Подробнее
7	Шабашкин Егор Игорьевич	2,0	2	Подробнее
8	Гирин Константин Денисович	1,0	1	Подробнее
9	Лампочкин Анатолий Валерьевич	0,0	0	Подробнее

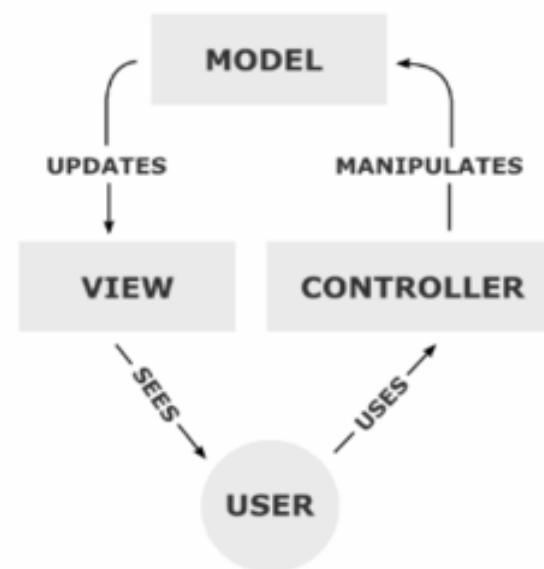
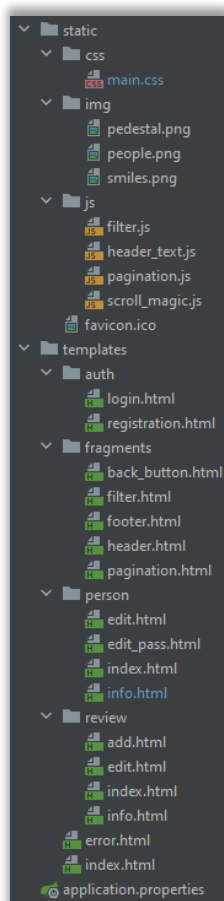
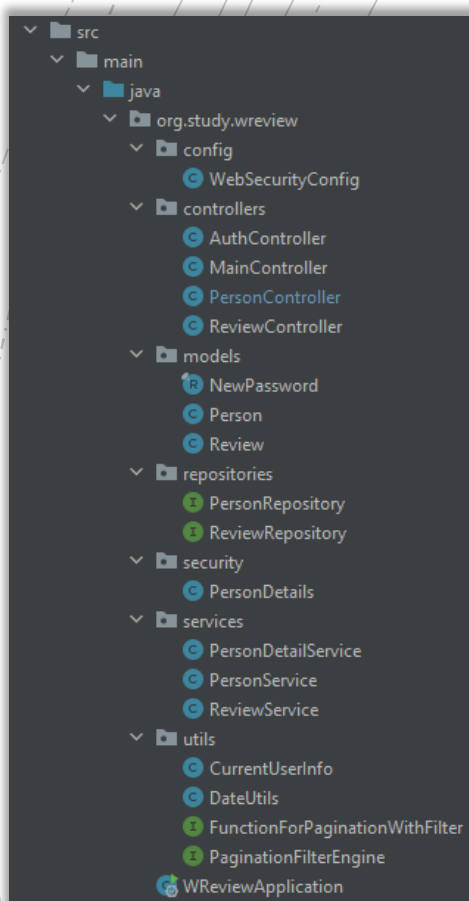


Используемые технологии

- Язык программирования Java
- Spring Framework:
 1. Core
 2. Spring boot
 3. Web
 4. Data JPA
 5. Security
- PostgreSQL
- Thymeleaf
- Lombok
- HTML, CSS, JavaScript

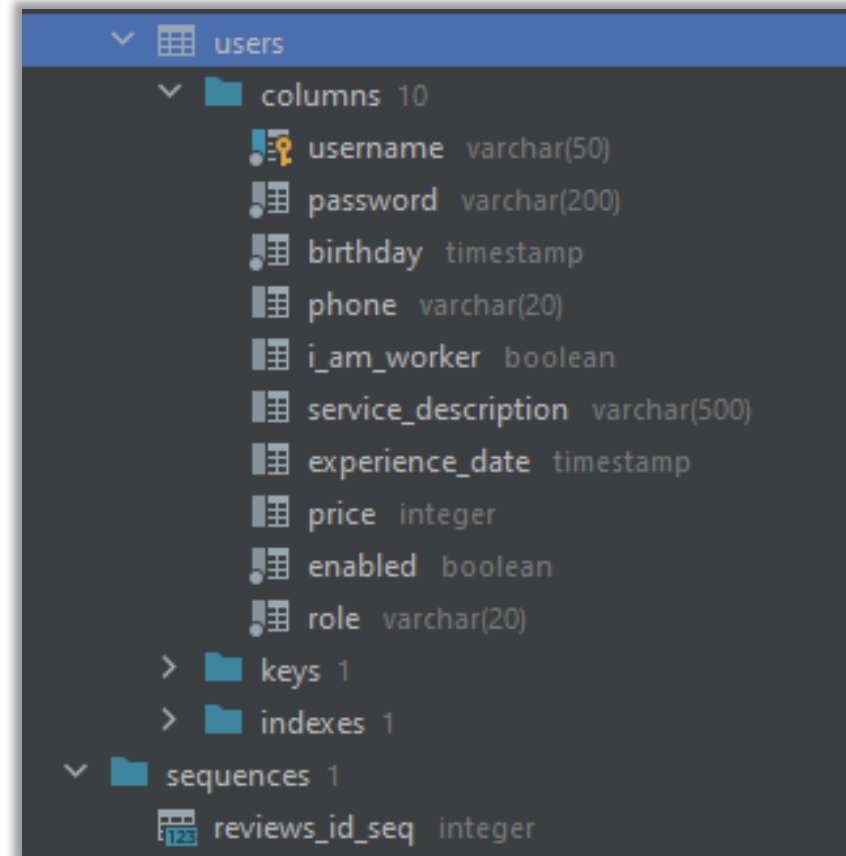
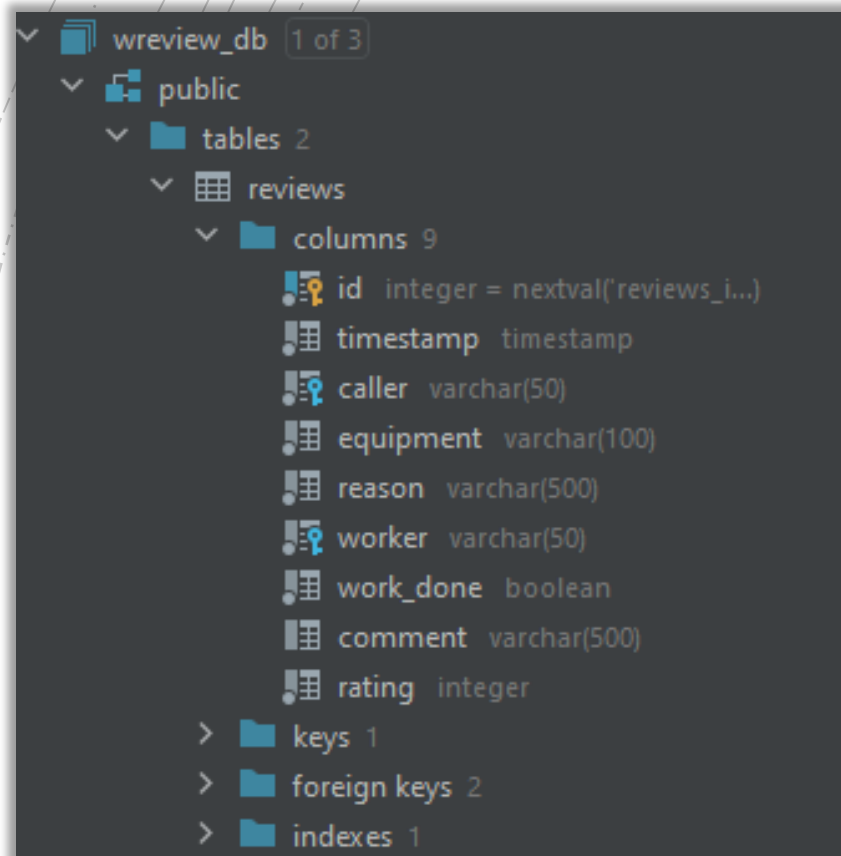


Структура проекта



MVC - схема разделения данных приложения и управляющей логики на три отдельных компонента: модель, представление и контроллер.

Структура базы данных



Немного кода

```
14  @Controller
15  @RequiredArgsConstructor
16  @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
17  public class MainController {
18
19      PersonService personService;
20
21      @GetMapping("")
22      String index(Model model){
23          List<Person> persons = personService.findWorkersWithRating();
24          model.addAttribute("persons", persons);
25          return "index";
26      }
27  }
```

Ещё немного кода

```
14 @Data
15 @Entity
16 @ToString
17 @NoArgsConstructor
18 @Table(name = "reviews")
19 @FieldDefaults(level = AccessLevel.PRIVATE)
20 public class Review {
21     @Id
22     @GeneratedValue(strategy = GenerationType.IDENTITY)
23     long id;
24
25     @Temporal(TemporalType.TIMESTAMP)
26     @DateTimeFormat(pattern = "dd.MM.yyyy HH:mm")
27     Date timestamp;
28
29     @ManyToOne
30     @JoinColumn(name = "caller", referencedColumnName = "username")
31     Person caller;
```

```
33 @NotEmpty(message = "Поле 'оборудование' не должно быть пустым")
34 @Size(max = 100, message = "Поле 'оборудование' должно содержать не более 100 символов")
35 String equipment;
36
37 @NotEmpty(message = "Поле 'причина вызова' не должно быть пустым")
38 @Size(max = 500, message = "Поле 'причина вызова' должно содержать не более 500 символов")
39 String reason;
40
41 @ManyToOne
42 @JoinColumn(name = "worker", referencedColumnName = "username")
43 Person worker;
44
45 @Column(name = "work_done")
46 boolean workDone;
47
48 @Size(max = 500, message = "Поле 'комментарий' должно содержать не более 500 символов")
49 String comment;
50
51 @Min(value = 0, message = "Оценка не должна быть ниже 0")
52 @Max(value = 10, message = "Оценка не должна быть выше 10")
53 @NotNull(message = "Это поле не должно быть пустым")
54 Integer rating;
```

Ещё немного кода

```
15 @Repository
16 public interface PersonRepository extends JpaRepository<Person, String> {
17
18     4 usages  🧑 Dmitry
19     Optional<Person> findByUsername(String username);
20
21     1 usage  🧑 Dmitry
22     Optional<Person> findByUsernameAndIamWorkerTrue(String username);
23
24     2 usages  🧑 Dmitry
25     List<Person> findByIamWorkerTrueOrderByUsername();
26
27     1 usage  🧑 Dmitry
28     @Query(value = "select * from users" +
29         " where username ~* :filter or phone ~* :filter or service_description ~* :filter",
30         nativeQuery = true)
31     Page<Person> findAllWithFilter(@Param("filter") String filter, Pageable pageable);
32
33     1 usage  🧑 Dmitry
34     @Query(value = "select * from users" +
35         " where i_am_worker = true and (username ~* :filter or phone ~* :filter or service_description ~* :filter)",
36         nativeQuery = true)
37     Page<Person> findWorkersWithFilter(@Param("filter") String filter, Pageable pageable);
38 }
```

Ещё немного кода

```
18 @Service
19 @RequiredArgsConstructor
20 @Transactional(readOnly = true)
21 @FieldDefaults(level = AccessLevel.PRIVATE, makeFinal = true)
22 public class ReviewService {
23     ReviewRepository reviewRepository;
24
25     no usages  Dmitry
26     public Page<Review> findAll(Pageable pageable) { return reviewRepository.findAll(pageable); }
27
28     1 usage  Dmitry
29     public Page<Review> findAllByCaller(String caller, String filter, Pageable pageable){
30         return reviewRepository.findAllByCaller(caller, filter, pageable);
31     }
32
33     1 usage  Dmitry
34     public Page<Review> findAllByWorker(String worker, String filter, Pageable pageable){
35         return reviewRepository.findAllByWorker(worker, filter, pageable);
36     }
37
38     1 usage  Dmitry
39     public Page<Review> findAllWithFilter(String filter, Pageable pageable){
40         return reviewRepository.findReviewsWithFilter(filter, pageable);
41     }
42
43     4 usages  Dmitry
44     public Optional<Review> findById(long id) { return reviewRepository.findById(id); }
```

```
41 4 usages  Dmitry
42     public Optional<Review> findById(long id) { return reviewRepository.findById(id); }
43
44
45     1 usage  Dmitry
46     @Transactional
47     public void save(Review review) { reviewRepository.save(review); }
48
49
50     1 usage  Dmitry
51     @Transactional
52     public void update(long id, Review review){
53         reviewRepository.findById(id).ifPresent(r -> {
54             r.setEquipment(review.getEquipment());
55             r.setReason(review.getReason());
56             r.setWorker(review.getWorker());
57             r.setWorkDone(review.isWorkDone());
58             r.setComment(review.getComment());
59             r.setRating(review.getRating());
60             reviewRepository.save(r);
61         });
62
63     1 usage  Dmitry
64     @Transactional
65     public void delete(Review review) { reviewRepository.delete(review); }
66
67 }
```

Ещё немного кода

```
15 @Configuration
16 @EnableWebSecurity
17 public class WebSecurityConfig {
18     @Bean
19     public SecurityFilterChain securityFilterChain(HttpSecurity http) throws Exception {
20         http
21             .authorizeHttpRequests(request -> request
22                 .requestMatchers("/persons", "/persons/*/lock").hasRole("ADMIN")
23                 .requestMatchers(HttpMethod.DELETE, "/persons/*").hasRole("ADMIN")
24                 .requestMatchers(HttpMethod.POST, "/persons/*").hasAnyRole("ADMIN", "USER")
25                 .requestMatchers(HttpMethod.PATCH, "/persons/*").hasAnyRole("ADMIN", "USER")
26                 .requestMatchers(HttpMethod.GET, "/persons/*").permitAll()
27                 .requestMatchers("/persons/**", "/reviews/**").hasAnyRole("ADMIN", "USER")
28                 .requestMatchers("/", "/login", "/logout", "/registration",
29                     "/css/*", "/js/*", "/img/*", "/favicon.ico").permitAll()
30             )
31             .formLogin(form -> form
32                 .loginPage("/login")
33                 .loginProcessingUrl("/process_login")
34                 .defaultSuccessUrl("/", alwaysUse: true)
35             )
36             .logout(logout -> logout
37                 .logoutSuccessUrl("/")
38                 .invalidateHttpSession(true)
39                 .deleteCookies("JSESSIONID"));
40         return http.build();
41     }
42 }
43 @Bean
44 public PasswordEncoder getPasswordEncoder() { return new BCryptPasswordEncoder( strength: 14); }
45 }
```

Заключение

Имея под рукой приложение WReview, не составит труда определить претендентов на трудовые награды. Труд каждого коллеги будет оценен!



Спасибо за внимание!



Автор: Демьяненко Дмитрий Сергеевич, инженер-электроник КБФГ, тел. 8-919-455-35-03