Technical University of Cluj-Napoca

OpenPick

Open Source Project Management Application

Software Engineering Laboratory Project

Name: Antonia Zară
Team: Antonia Zară and Doroteea Șandor
Group: 30433
Email: zaraantonia@yahoo.com

Contents

1 Abstract

OpenPick is a Java Spring application meant to aid students and IT enthusiasts into participating in open-source projects. The idea behind the application is that the user can create a profile, post their own open source project along with tasks and needed documents and wait for other users to take matter into their own hands and start editing the documents. Alternatively, an user can pick a project that most represents them and can start solving the written tasks.

2 Functionalities

The user can edit their account. They can register, login and logout. Additionally, for the projects that they own (meaning that they have created), they can edit and delete them, as well as add, delete, and edit their tasks. For other projects, they can edit the documents.

Each user can see the projects page, which should all the projects in the application, and their profile section.

The admin can delete projects and users, acting as a moderator for the wellbeing of the application. File size for the documents is of maximum 20MB.

3 Technologies Used

The application has been written in Java Spring as for the back-end. For security protocols we have used Spring Security to ensure that the login/logout/register operations do not allow interference to the database. The front-end of the applications uses Bootstrap, Thymeleaf as well as a lot of plain HTML and CSS code for the aesthetics of our project. The database management is implemented through the frameworks JPA and Hibernate, for a better and safer data access.

4 Design Pattern and Architecture

The design pattern chosen for the project is the Model View Controller pattern, specifically preferred by the authors of Java Spring for their framework. The architecture chosen is a five layer architecture mainly centered around the database access, where we implement repository classes as data access objects, service classes as business layers to handle the repositories, model classes for modelling the objects in our application, controllers for interaction between the front end and back end and view for the front end.

5 Project Design

5.1 Use Case Diagram (Figure 1)

5.2 Class Diagram (Figure 2)

5.3 Deployment Diagram (Figure 3)

6 Project Implementation

All implementations can be found in the listing in the appendix. Back end is in listing 1-17, and front end is in listings starting with 18.

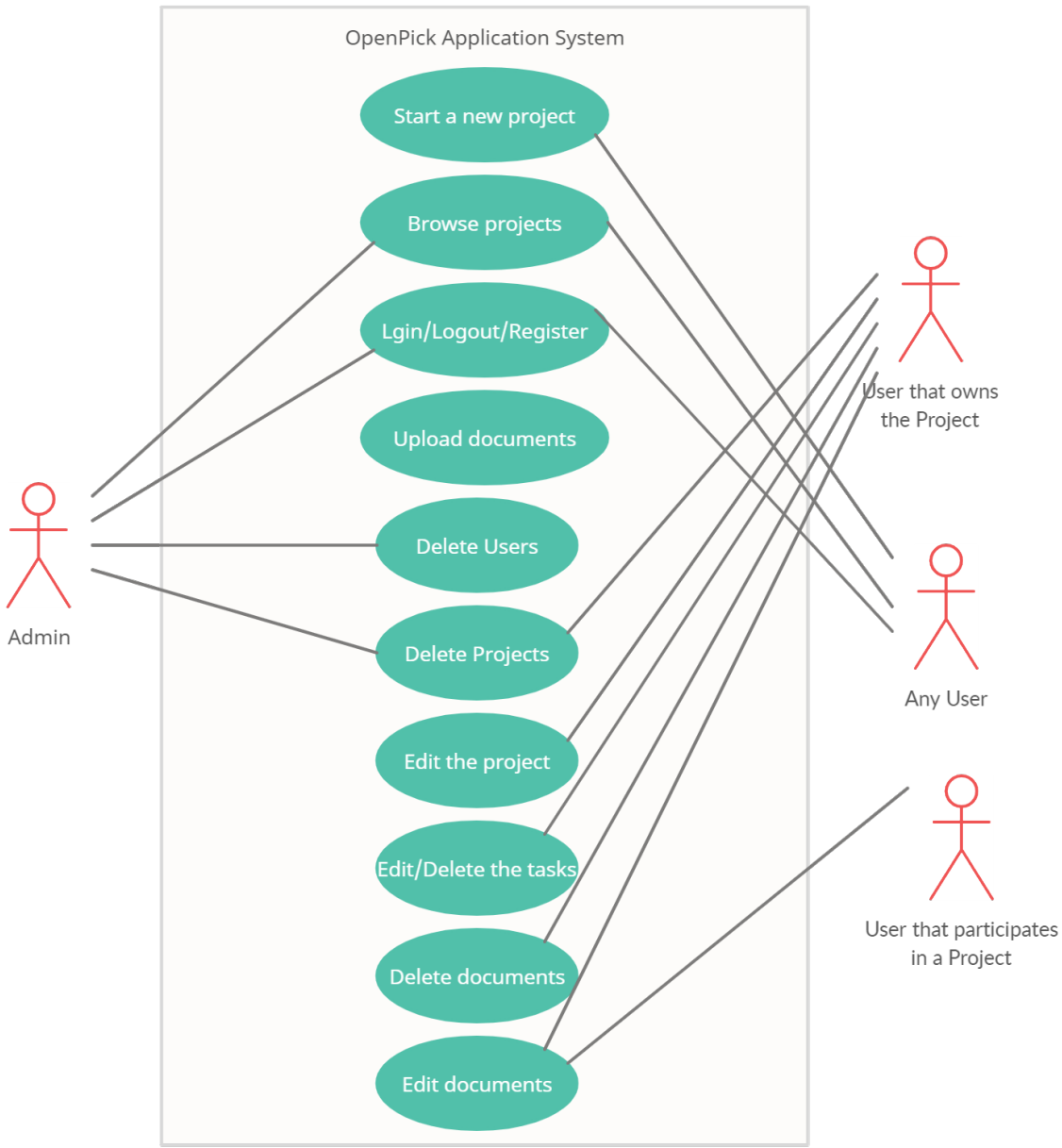

Figure 1: Use Case Diagram.

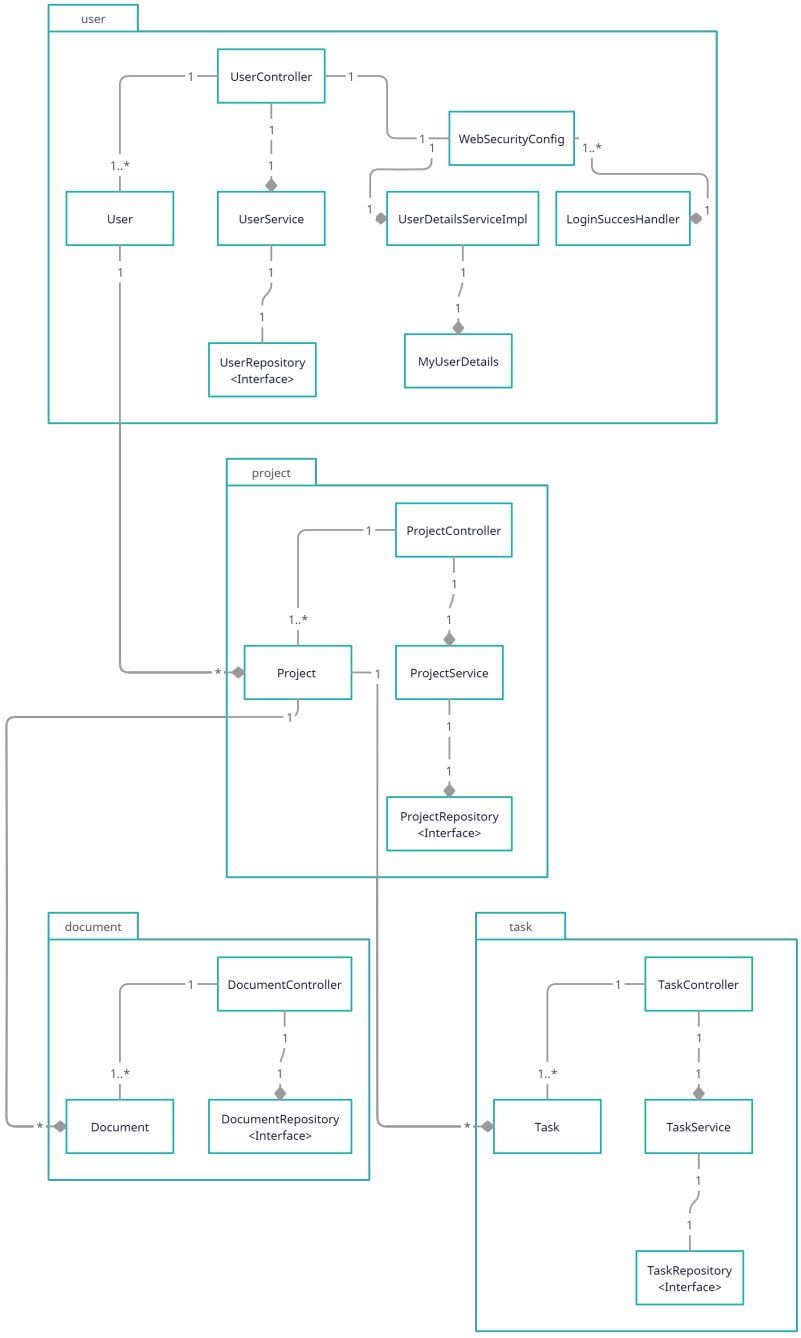

Figure 2: Class Diagram.

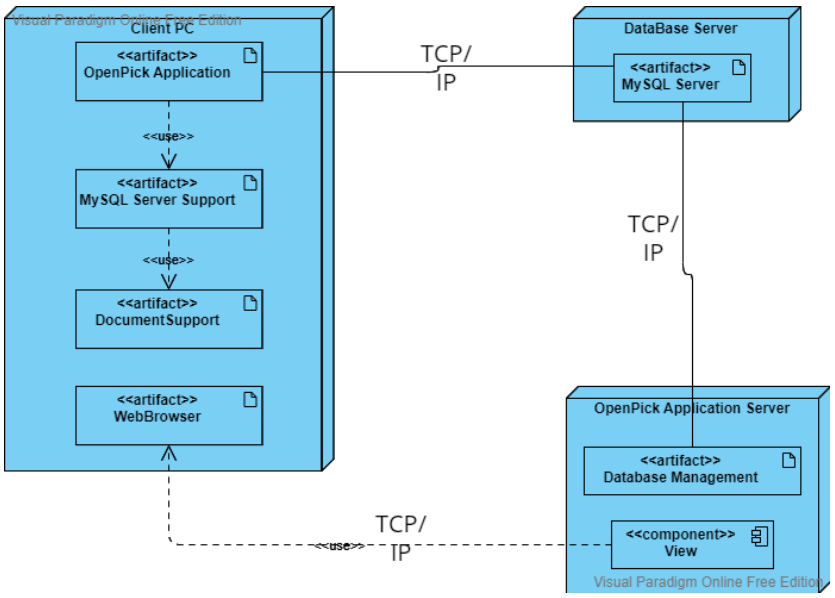

Figure 3: Deployment Diagram.

```java
package com.softwareeng.openpick.user;


import com.softwareeng.openpick.project.Project;

import javax.persistence.*;
import java.util.List;

@Entity
@Table(name = "users")
public class User {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Integer id;

    @Column(nullable = false, unique = true, length = 45)
    private String email;

    @Column(length = 100, nullable = false) //name is default here
    private String password;

    @Column(length = 45, nullable = false)
    private  String username;

    @Column(length=12,nullable = false)
    private String role;

    private boolean enabled;


    public String getRole() {
        return role;
    }

    public void setRole(String role) {
        this.role = role;
    }

    @OneToMany(mappedBy = "owner", cascade = CascadeType.ALL, orphanRemoval = false)
    private List<Project> projects;


    public Integer getId() {
```

```
        return id;
    }

    public void setId(Integer id) {
        this.id = id;
    }

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }

    public String getUsername() {
        return username;
    }

    public void setUsername(String username) {
        this.username = username;
    }

    @Override
    public String toString() {
        return "User{" +
                "id=" + id +
                ", email='" + email + '\'' +
                ", password='" + password + '\'' +
                ", username='" + username + '\'' +
                ", enabled=" + enabled +
                '}';
    }

    public boolean isEnabled() {
        return enabled;
    }

    public void setEnabled(boolean enabled) {
        this.enabled = enabled;
    }

    public List<Project> getProjects() {
        return projects;
    }

    public void setProjects(List<Project> projects) {
        if(this.projects == null){
            this.projects = projects;
        }
        else{
            this.projects.clear();
            this.projects.addAll(projects);
        }
    }
}
```

Listing 1: User.java

```
package com.softwareeng.openpick.user;


import com.softwareeng.openpick.exception.NotFoundException;
```

```java
import com.softwareeng.openpick.project.ProjectService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.security.core.context.SecurityContextHolder;
import org.springframework.security.core.userdetails.UserDetails;
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.*;
import org.springframework.web.servlet.mvc.support.RedirectAttributes;
import java.util.List;

@Controller
public class UserController {
    @Autowired
    private UserService service;

    @Autowired
    private ProjectService projectService;

    @GetMapping("/users")
    public String showUserList(Model model) {
        List<User> listUsers = service.listAll();
        model.addAttribute("listUsers", listUsers);
        Object principal = SecurityContextHolder.getContext().getAuthentication().
    getPrincipal();

        if (principal instanceof UserDetails) {
            User loggedInUser = service.findByUsername(((UserDetails)principal).
    getUsername());
            model.addAttribute("loggedInUserId", loggedInUser.getId().toString());
            System.out.println(loggedInUser);
        } else {
            model.addAttribute("loggedInUserId", "");
        }

        return "users";
    }

    @GetMapping("/users/new")
    public String showNewForm(Model model) {
        model.addAttribute("user", new User());
        model.addAttribute("pageTitle", "Add New User");
        model.addAttribute("oldId", 0);
        return "user_form";
    }

    @PostMapping("/users/save")
    public String saveUser(User user, RedirectAttributes ra) throws
    UserNotFoundException {
        user.setRole("USER");
        service.save(user);
        return "redirect:/users/{oldId}";
    }

    @RequestMapping(value = "/users/save/{oldId}", method = RequestMethod.POST)
    public String saveUserAgain(@PathVariable("oldId") String oldId, Model model, User
     user, RedirectAttributes ra) {
        BCryptPasswordEncoder passwordEncoder = new BCryptPasswordEncoder();
        String encodedPassword = passwordEncoder.encode(user.getPassword());
        user.setPassword(encodedPassword);
        user.setId(Integer.parseInt(oldId));
        user.setRole("USER");
        service.save(user);

        ra.addFlashAttribute("message", "The user has been saved successfully.");
        return "redirect:/users/{oldId}";
    }

    @GetMapping("/users/edit/{id}")
    public String showEditForm(@PathVariable("id") Integer userId, Model model,
    RedirectAttributes ra) {
```

```
        try {
            User user = service.get(userId);
            model.addAttribute("user", user);
            model.addAttribute("pageTitle", "Edit User (ID: " + userId + ")");
            model.addAttribute("oldId", userId.toString());
            return "user_form";
        } catch (NotFoundException e) {
            ra.addFlashAttribute("message", "The .");
            return "redirect:/projects";
        }
    }

    @GetMapping("/users/delete/{id}")
    public String deleteUser(@PathVariable("id") Integer id, RedirectAttributes ra) {
        try {
            service.delete(id);
            ra.addFlashAttribute("message","User has been deleted succesfully");
        } catch (NotFoundException e) {
            ra.addFlashAttribute("message", e.getMessage());
        }
        return "redirect:/users";
    }


    @GetMapping("/register")
    public String showRegistrationForm(Model model) {
        model.addAttribute("user", new User());

        return "signup_form";
    }

    @PostMapping("/process_register")
    public String processRegister(User user) {
        BCryptPasswordEncoder passwordEncoder = new BCryptPasswordEncoder();
        String encodedPassword = passwordEncoder.encode(user.getPassword());
        user.setPassword(encodedPassword);
        user.setRole("USER");
        user.setEnabled(true);
        service.save(user);

        return "register_success";
    }

    @GetMapping("/users/{id}")
    public String viewProfileOfUser(Model model, @PathVariable("id") Integer id,
    RedirectAttributes ra) {
        try {
            User currentUser = service.get(id);
            model.addAttribute("currentUser", currentUser);
            ra.addFlashAttribute("message","User has been deleted succesfully");
        } catch (NotFoundException e) {
            ra.addFlashAttribute("message", e.getMessage());
        }
        return "profile";
    }

}
```

Listing 2: UserController.java

```
package com.softwareeng.openpick.user;

public class UserNotFoundException extends Throwable {
    public UserNotFoundException(String message) {
        super(message);
    }
}
```

Listing 3: UserNotFoundException.java

```
package com.softwareeng.openpick.user;
```

```java

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.Modifying;
import org.springframework.data.jpa.repository.Query;
import org.springframework.data.repository.CrudRepository;
import org.springframework.data.repository.query.Param;
import org.springframework.stereotype.Repository;
import org.springframework.transaction.annotation.Transactional;

@Repository
public interface UserRepository extends JpaRepository<User, Integer> {
    public Long countById(Integer id);

    @Query("SELECT u FROM User u WHERE u.username = :username")
    public User getUserByUsername(@Param("username") String username);

    @Transactional
    @Modifying
    @Query(value = "UPDATE users SET id = :newId WHERE id = :oldId", nativeQuery =
    true)
    public void updateUserId(Integer oldId, Integer newId);
}
```

Listing 4: UserRepository.java

```java
package com.softwareeng.openpick.user;

import com.softwareeng.openpick.exception.NotFoundException;
import com.softwareeng.openpick.project.ProjectService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.util.List;
import java.util.Optional;

@Service
public class UserService {
    @Autowired
    private UserRepository repo;

    @Autowired
    private ProjectService projectService;

    public List<User> listAll(){
        return (List<User>) repo.findAll();
    }

    public void save(User user){
        repo.save(user);
    }

    public User get(Integer id) throws NotFoundException {
        Optional<User> result = repo.findById(id);
        if (result.isPresent()) {
            User currentUser = result.get();
            return currentUser;
        }
        throw new NotFoundException("Could not find any users with ID " + id);
    }

    public void delete(Integer id) throws NotFoundException {
        Long count = repo.countById(id);
        if(count == null | count == 0){
            throw new NotFoundException("Could not find user.");
        }
        repo.deleteById(id);
    }

    public User findByUsername(String username) {
        return repo.getUserByUsername(username);
    }
```

```
}
```

Listing 5: UserService.java

```
package com.softwareeng.openpick.task;

import com.softwareeng.openpick.project.Project;

import javax.persistence.*;
import java.sql.Date;

@Entity
@Table(name = "tasks")
public class Task {


    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Integer id;

    @Column(nullable = false, name = "title", length = 25)
    private String title;

    @Column(nullable = false, name = "start_date")
    private Date startDate;

    @Column(nullable = false, name = "deadline")
    private Date deadline;

    @Column(name = "description", length = 255)
    private String description;

    @ManyToOne
    @JoinColumn(name = "project_id", referencedColumnName = "id")
    private Project project;


    public void setId(Integer id) {
        this.id = id;
    }

    public Integer getId() {
        return id;
    }

    public String getTitle() {
        return title;
    }

    public void setTitle(String title) {
        this.title = title;
    }

    public Date getStartDate() {
        return startDate;
    }

    public void setStartDate(Date startDate) {
        this.startDate = startDate;
    }

    public Date getDeadline() {
        return deadline;
    }

    public void setDeadline(Date deadline) {
        this.deadline = deadline;
    }

    public Project getProject() {
        return project;
```

```
    }

    public void setProject(Project project) {
        this.project = project;
    }

    public String getDescription() {
        return description;
    }

    public void setDescription(String description) {
        this.description = description;
    }


}
```

Listing 6: Task.java

```
package com.softwareeng.openpick.task;

import com.softwareeng.openpick.exception.NotFoundException;
import com.softwareeng.openpick.project.ProjectService;
import com.softwareeng.openpick.user.UserService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.servlet.mvc.support.RedirectAttributes;

@Controller
public class TaskController {

    @Autowired
    private TaskService taskService;

    @Autowired
    private UserService userService;

    @Autowired
    private ProjectService projectService;

    @GetMapping("/users/{user_id}/projects/{project_id}/newtask")
    public String addTask(@PathVariable("user_id") Integer userId, @PathVariable("
    project_id") Integer projectId, Model model, RedirectAttributes ra) {
        try {
            model.addAttribute("task", new Task());
            model.addAttribute("currentUser", userService.get(userId));
            model.addAttribute("currentProject", projectService.get(projectId));
            model.addAttribute("pageTitle", "Add Task");
        } catch (NotFoundException e) {
            return "redirect:/users/{user_id}/projects/{project_id}";
        }

        ra.addFlashAttribute("message", "User has been deleted succesfully");
        return "task_form";
    }

    @RequestMapping("/users/{user_id}/projects/{project_id}/savetask")
    public String saveTask(@PathVariable("user_id") Integer userId, @PathVariable("
    project_id") Integer projectId, Task task, RedirectAttributes ra) {
        try {
            task.setProject(projectService.get(projectId));
            taskService.save(task);
            ra.addFlashAttribute("message", "The task has been saved successfully.");
        } catch (NotFoundException e) {
            ra.addFlashAttribute("message", "The task saved unsuccessfully.");
        }
        return "redirect:/users/{user_id}/projects/{project_id}";
```

```
    }

    @GetMapping("/users/{user_id}/projects/{project_id}/deletetask/{task_id}")
    public String deleteTask(@PathVariable("task_id") Integer taskId,
    RedirectAttributes ra) {
        try {
            taskService.delete(taskService.get(taskId));
        } catch (NotFoundException e) {
            ra.addFlashAttribute("message", "The task deleted unsuccessfully.");
        }
        return "redirect:/users/{user_id}/projects/{project_id}";
    }

    @GetMapping("/users/{user_id}/projects/{project_id}/edittask/{task_id}")
    public String showEditFormTask(@PathVariable("user_id") Integer userId,
    @PathVariable("project_id") Integer projectId,@PathVariable("task_id") Integer
    taskId, Model model, RedirectAttributes ra) {
        try {
            model.addAttribute("task", taskService.get(taskId));
            model.addAttribute("currentUser", userService.get(userId));
            model.addAttribute("currentProject", projectService.get(projectId));
            model.addAttribute("pageTitle", "Edit Task");
            return "task_form_edit";
        } catch (NotFoundException e) {
            ra.addFlashAttribute("message", "The task deleted unsuccessfully.");
        }
        return "redirect:/users/{user_id}/projects/{project_id}";
    }

    @RequestMapping("/users/{user_id}/projects/{project_id}/savetask/{task_id}")
    public String editTask(@PathVariable("task_id") Integer taskId, @PathVariable("
    project_id") Integer projectId, Task task, RedirectAttributes ra) {
        try {
            task.setId(taskId);
            task.setProject(projectService.get(projectId));
            taskService.save(task);
            ra.addFlashAttribute("message", "The task has been modified successfully."
    );
        } catch (NotFoundException e) {
            ra.addFlashAttribute("message", "The task has been modified unsuccessfully
    .");
        }
        return "redirect:/users/{user_id}/projects/{project_id}";
    }

}
```

Listing 7: TaskController.java

```
package com.softwareeng.openpick.task;

import org.springframework.data.repository.CrudRepository;

public interface TaskRepository extends CrudRepository<Task,Integer> {
}
```

Listing 8: TaskRepository.java

```
package com.softwareeng.openpick.task;

import com.softwareeng.openpick.exception.NotFoundException;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.util.Optional;

@Service
public class TaskService {

    @Autowired
    private TaskRepository repo;
```

```
14
15      public void save(Task task) {
16          repo.save(task);
17      }
18
19      public Task get(Integer taskId) throws NotFoundException {
20          Optional<Task> result = repo.findById(taskId);
21          if (result.isPresent()){
22              return result.get();
23          }
24          throw new NotFoundException("Could not find any tasks with ID " + taskId);
25      }
26
27      public void delete(Task task) {
28          repo.delete(task);
29      }
30  }
```

Listing 9: TaskService.java

```
1   package com.softwareeng.openpick.project;
2
3   import com.softwareeng.openpick.document.Document;
4   import com.softwareeng.openpick.task.Task;
5   import com.softwareeng.openpick.user.User;
6
7   import javax.persistence.*;
8   import java.util.ArrayList;
9   import java.util.List;
10
11  @Entity
12  @Table(name = "projects")
13  public class Project {
14
15      @Id
16      @GeneratedValue(strategy = GenerationType.IDENTITY)
17      private Integer id;
18
19      @Column(nullable = false, length = 25)
20      private String title;
21
22      @Column(nullable = false, length = 255)
23      private String description;
24
25      @ManyToOne
26      @JoinColumn(name = "user_id", referencedColumnName = "id")
27      private User owner;
28
29      public Project(String title, String description, User user) {
30          this.title = title;
31          this.description = description;
32          this.owner = user;
33      }
34
35      public Project() {
36
37      }
38
39      @OneToMany(mappedBy = "project", cascade = CascadeType.ALL, orphanRemoval = false)
40      private List<Task> tasks = new ArrayList<>();
41
42      @OneToMany(mappedBy = "project", cascade = CascadeType.ALL, orphanRemoval = false)
43      private List<Document> documents = new ArrayList<>();
44
45      public Integer getId() {
46          return id;
47      }
48
49      public void setId(Integer id) {
50          this.id = id;
51      }
```

```
52
53      public String getTitle() {
54          return title;
55      }
56
57      public void setTitle(String title) {
58          this.title = title;
59      }
60
61      public User getOwner() {
62          return owner;
63      }
64
65      public void setOwner(User owner) {
66          this.owner = owner;
67      }
68
69      public List<Task> getTasks() {
70          return tasks;
71      }
72
73      public void setTasks(List<Task> tasks) {
74          this.tasks = tasks;
75      }
76
77      public String getDescription() {
78          return description;
79      }
80
81      public void setDescription(String description) {
82          this.description = description;
83      }
84
85      public List<Document> getDocuments() {
86          return documents;
87      }
88
89      public void setDocuments(List<Document> documents) {
90          this.documents = documents;
91      }
92
93      @Override
94      public String toString() {
95          return "Project{" +
96                  "id=" + id +
97                  ", title='" + title + '\'' +
98                  ", description='" + description + '\'' +
99                  ", owner=" + owner +
100                 '}';
101     }
102 }
```

Listing 10: Project.java

```
1  package com.softwareeng.openpick.project;
2
3  import com.softwareeng.openpick.exception.NotFoundException;
4  import com.softwareeng.openpick.user.User;
5  import com.softwareeng.openpick.user.UserNotFoundException;
6  import com.softwareeng.openpick.user.UserService;
7  import org.springframework.beans.factory.annotation.Autowired;
8  import org.springframework.security.core.context.SecurityContextHolder;
9  import org.springframework.security.core.userdetails.UserDetails;
10 import org.springframework.stereotype.Controller;
11 import org.springframework.ui.Model;
12 import org.springframework.web.bind.annotation.GetMapping;
13 import org.springframework.web.bind.annotation.PathVariable;
14 import org.springframework.web.bind.annotation.RequestMapping;
15 import org.springframework.web.servlet.mvc.support.RedirectAttributes;
16
17 import java.util.List;
```

```
18
19 @Controller
20 public class ProjectController {
21
22     @Autowired
23     private ProjectService service;
24
25     @Autowired
26     private UserService userService;
27
28     @GetMapping("/projects")
29     public String showAllProjects(Model model){
30         List<Project> allProjects = service.findAll();
31         model.addAttribute("allProjects", allProjects);
32         Object principal = SecurityContextHolder.getContext().getAuthentication().
       getPrincipal();
33
34         if (principal instanceof UserDetails) {
35
36             User loggedInUser = userService.findByUsername(((UserDetails)principal).
       getUsername());
37             model.addAttribute("loggedInUserId", loggedInUser.getId().toString());
38
39             if(loggedInUser.getRole().equals("ADMIN")){
40                 return "all_projects_admin";
41             }
42
43         } else {
44             model.addAttribute("loggedInUserId", "");
45         }
46         return "all_projects";
47     }
48
49
50     @GetMapping("/users/{user_id}/projects/{project_id}")
51     public String showProjectsFromUser(@PathVariable("user_id") Integer userId,
       @PathVariable("project_id") Integer projectId, Model model, RedirectAttributes ra){
52         try{
53             Project project = service.get(projectId);
54             model.addAttribute("currentProject",project);
55             if(project.getTitle() == null) throw new NotFoundException("ew");
56             model.addAttribute("currentUser", userService.get(userId));
57
58             Object principal = SecurityContextHolder.getContext().getAuthentication().
       getPrincipal();
59
60             if (principal instanceof UserDetails) {
61                 User loggedInUser = userService.findByUsername(((UserDetails)principal
       ).getUsername());
62                 model.addAttribute("loggedInUserId", loggedInUser.getId().toString());
63
64                 if(userId.equals(loggedInUser.getId())){
65                     return "project_view_owner";
66                 }
67                 else{
68                     return "project_view_other";
69                 }
70
71             } else {
72                 model.addAttribute("loggedInUserId", "");
73                 return "project_view_other";
74             }
75         }
76         catch (NotFoundException e) {
77             return "/users/{user_id}";
78         }
79     }
80
81     @GetMapping("/users/{user_id}/projects/new")
82     public String addNewProjectToUser(@PathVariable("user_id") Integer userId, Model
       model){
```

```
        try {
            User currentUser = userService.get(userId);
            model.addAttribute("currentUser", currentUser);
            model.addAttribute("projectt", new Project());
            return "project_form";
        } catch (NotFoundException e) {
            return "/users/{user_id}";
        }
    }

    @GetMapping("/users/{user_id}/projects/{project_id}/edit")
    public String showEditFormProject(@PathVariable("user_id") Integer userId,
    @PathVariable("project_id") Integer projectId, Model model, RedirectAttributes ra)
    {
        try {
            Project project = service.get(projectId);
            model.addAttribute("ourProject", project);
            model.addAttribute("pageTitle", "Edit Project");
            model.addAttribute("userId", userId.toString());
            model.addAttribute("projectId", projectId.toString());
            return "project_form_edit";
        } catch (NotFoundException e) {
            ra.addFlashAttribute("message", "The .");
            return "redirect:/users";
        }
    }

    @RequestMapping("/users/{user_id}/projects/save")
    public String saveProject(@PathVariable("user_id") Integer userId, Project
    ourProject, RedirectAttributes ra, Model model) throws UserNotFoundException,
    NotFoundException {
        ourProject.setOwner(userService.get(userId));
        service.save(ourProject);

        ra.addFlashAttribute("message", "The project has been saved successfully.");
        return "redirect:/users/{user_id}";
    }

    @RequestMapping("/users/{user_id}/projects/{project_id}/save")
    public String saveProject(@PathVariable("user_id") Integer userId, @PathVariable("
    project_id") Integer projectId, Project ourProject, RedirectAttributes ra, Model
    model) throws UserNotFoundException, NotFoundException {
        ourProject.setOwner(userService.get(userId));
        service.save(ourProject);

        ra.addFlashAttribute("message", "The project has been saved successfully.");
        return "redirect:/users/{user_id}";
    }

    @GetMapping("/users/{user_id}/projects/{project_id}/delete")
    public String deleteProject(Model model, @PathVariable("user_id") Integer userId,
    @PathVariable("project_id") Integer projectId, RedirectAttributes ra) {
        service.delete(projectId);

        ra.addFlashAttribute("message","Project has been deleted succesfully");

        Object principal = SecurityContextHolder.getContext().getAuthentication().
    getPrincipal();

        if (principal instanceof UserDetails) {

            User loggedInUser = userService.findByUsername(((UserDetails)principal).
    getUsername());
            model.addAttribute("loggedInUserId", loggedInUser.getId().toString());

            if(loggedInUser.getRole().equals("ADMIN")){
                return "redirect:/projects";
            }
        } else {
            model.addAttribute("loggedInUserId", "");
        }
```

```
        return "redirect:/users/{user_id}";
    }

}
```

Listing 11: ProjectController.java

```
package com.softwareeng.openpick.project;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.Modifying;
import org.springframework.data.jpa.repository.Query;
import org.springframework.data.repository.query.Param;
import org.springframework.transaction.annotation.Transactional;

import java.util.List;

public interface ProjectRepository extends JpaRepository<Project,Integer> {

    @Query(value = "SELECT projects_id FROM users_projects WHERE user_id = :userId",
    nativeQuery = true)
    public List<Integer> findProjectsIdByUserId(@Param("userId") Integer userId);

    @Transactional
    @Modifying
    @Query(value = "UPDATE users_projects SET user_id = :newId WHERE user_id = :oldId"
    , nativeQuery = true)
    public void updateUserId(@Param("oldId") Integer oldId, @Param("newId")Integer
    newId);

    @Transactional
    @Modifying
    @Query(value = "UPDATE projects SET user_id = :newId WHERE user_id = :oldId",
    nativeQuery = true)
    public void updateUserIdInProject(Integer oldId, Integer newId);

    @Transactional
    @Modifying
    @Query(value = "INSERT INTO users_projects (user_id, projects_id) VALUES (:userId,
     :projectId)", nativeQuery = true)
    void saveInUsersProjects(@Param("projectId") Integer projectId, @Param("userId")
    Integer userId);

    @Transactional
    @Modifying
    @Query(value = "DELETE FROM users_projects WHERE projects_id = :projectId",
    nativeQuery = true)
    void deleteProjectsFromUserProjects(@Param("projectId") Integer projectId);
}
```

Listing 12: ProjectRepository.java

```
package com.softwareeng.openpick.project;

import com.softwareeng.openpick.exception.NotFoundException;
import com.softwareeng.openpick.user.User;
import org.springframework.beans.factory.annotation.Autowired;
//import org.springframework.data.jpa.repository.Query;
import org.springframework.stereotype.Service;
//import org.springframework.data.repository.query.Param;
//import org.springframework.transaction.annotation.Transactional;


import java.util.List;
import java.util.Optional;

@Service
public class ProjectService {

    @Autowired
    private ProjectRepository repo;
```

```
20
21      public Project get(Integer projectId) throws NotFoundException {
22          Optional<Project> result = repo.findById(projectId);
23          if (result.isPresent()){
24              return result.get();
25          }
26          throw new NotFoundException("Could not find any projects with ID " + projectId
        );
27      }
28
29      public void updateUserIdInProject(Integer oldId, Integer newId) {
30          repo.updateUserIdInProject(oldId,newId);
31      }
32
33      public void save(Project project) {
34          repo.save(project);
35      }
36
37      public void saveInUsersProjects(Integer pid, Integer uid) {
38          repo.saveInUsersProjects(pid, uid);
39      }
40
41      public void delete(Integer id) {
42          repo.delete(repo.getById(id));
43      }
44
45      public List<Project> findAll() {
46          return repo.findAll();
47      }
48  }
```

Listing 13: ProjectService.java

```
1  package com.softwareeng.openpick.exception;
2
3  public class NotFoundException extends Throwable {
4      public NotFoundException(String message) {
5          super(message);
6      }
7  }
```

Listing 14: NotFoundException.java

```
1  package com.softwareeng.openpick.document;
2
3  import com.softwareeng.openpick.exception.NotFoundException;
4  import com.softwareeng.openpick.project.ProjectService;
5  import com.softwareeng.openpick.user.UserNotFoundException;
6  import org.springframework.beans.factory.annotation.Autowired;
7  import org.springframework.data.repository.query.Param;
8  import org.springframework.stereotype.Controller;
9  import org.springframework.ui.Model;
10 import org.springframework.util.StringUtils;
11 import org.springframework.web.bind.annotation.*;
12 import org.springframework.web.multipart.MultipartFile;
13 import org.springframework.web.servlet.mvc.support.RedirectAttributes;
14 import javax.servlet.ServletOutputStream;
15 import javax.servlet.http.HttpServletResponse;
16 import java.io.IOException;
17 import java.util.Date;
18 import java.util.List;
19 import java.util.Optional;
20
21
22 @Controller
23 public class DocumentController {
24
25     @Autowired
26     private DocumentRepository repo;
27
28     @Autowired
```

```
29      private ProjectService projectService;
30
31      @GetMapping("/documents")
32      public String showAllDocuments(Model model){
33          List<Document> listDocs = repo.findAll();
34          model.addAttribute("listDocs",listDocs);
35          return "documents";
36      }
37
38      @PostMapping("/upload/{project_id}")
39      public String uploadFile(@RequestParam("document") MultipartFile multipartFile,
        @PathVariable("project_id") Integer projectId, RedirectAttributes
        redirectAttributes) throws IOException {
40          String filename = StringUtils.cleanPath(multipartFile.getOriginalFilename());
41          Document document = new Document();
42          document.setContent(multipartFile.getBytes());
43          document.setSize(multipartFile.getSize());
44          document.setUploadTime(new Date());
45          document.setTitle(filename);
46          try {
47              document.setProject(projectService.get(projectId));
48              repo.save(document);
49              redirectAttributes.addFlashAttribute("message","The file has been uploaded
         successfully.");
50          } catch (NotFoundException ignored){}
51          return "redirect:/projects";
52      }
53
54      @GetMapping("/download")
55      public void downloadFile(@Param("id") Integer id, HttpServletResponse response )
        throws Exception {
56          Optional<Document> result = repo.findById(id);
57          if (!result.isPresent()) {
58              throw new Exception("Could not find document with ID:" + id);
59          }
60          Document document = result.get();
61          response.setContentType("application/octet-stream");
62          String headerKey = "Content-Disposition";
63          String headerValue = "attachment; filename=" + document.getTitle();
64
65          response.setHeader(headerKey, headerValue);
66
67          ServletOutputStream outputStream = response.getOutputStream();
68
69          outputStream.write(document.getContent());
70          outputStream.close();
71      }
72
73      @GetMapping("/delete/{id}")
74      public String deleteDocument(@PathVariable("id") Integer id, RedirectAttributes ra
        ) throws UserNotFoundException {
75          Long count = repo.countById(id);
76          if(count == null | count == 0){
77              throw new UserNotFoundException("Could not find document.");
78          }
79          repo.deleteById(id);
80          ra.addFlashAttribute("message","File has been deleted succesfully");
81          return "redirect:/projects";
82      }
83
84      @RequestMapping(value = "/uploadEdit/{oldId}", method = RequestMethod.POST)
85      public String editDocument(@PathVariable("oldId") Integer oldId, @RequestParam("
        document") MultipartFile multipartFile, RedirectAttributes redirectAttributes) {
86          try {
87              Optional<Document> result =repo.findById(oldId);
88              if(!result.isPresent()){
89                  throw new Exception("Could not find document with ID:"+oldId);
90              }
91              Document document = result.get();
92              String filename = StringUtils.cleanPath(multipartFile.getOriginalFilename
        ());
```

```
            document.setContent(multipartFile.getBytes());
            document.setSize(multipartFile.getSize());
            document.setUploadTime(new Date());
            document.setTitle(filename);
            repo.save(document);
            redirectAttributes.addFlashAttribute("message","The file has been edited
    successfully.");
            return "redirect:/projects";
        } catch (Exception e) {
            redirectAttributes.addFlashAttribute("message", "The .");
            return "redirect:/projects";
        }
    }

    @GetMapping("/edit/{id}")
    public String editDoc(@PathVariable("id") Integer id,Model model) {
        model.addAttribute("oldId", id.toString());
        return "doc_form";
    }

}
```

Listing 15: DocumentController.java

```
package com.softwareeng.openpick;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;

@Controller
public class MainController {

    @GetMapping("")
    public String showHomePage(){
        return "index";
    }

    @GetMapping("/about")
    public String showAboutPage(){
        return "about";
    }
}
```

Listing 16: MainController.java

```
package com.softwareeng.openpick;

import com.softwareeng.openpick.user.User;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class OpenPickApplication {

    public User loggedUser;

    public static void main(String[] args) {
        SpringApplication.run(OpenPickApplication.class, args);
    }
}
```

Listing 17: OpenPickApplication.java

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>About</title>
    <link rel="stylesheet" type="text/css" href="/css/style.css">
</head>
<body style="background: url(/images/shrek_girl.png);
```

```
    background-size: cover;
    background-repeat: no-repeat">

<!-- header -->
<div class="header">
    <div class="inner_header">
        <div class="logo_container">
            <h1>OPEN<span>PICK</span></h1>
        </div>

        <ul class="navigation">
            <a href="/"><li>Home</li></a>
            <a href="/about"><li>About</li></a>
            <a href="/logout"><li>Logout</li></a>
        </ul>

    </div>
</div>

<div style="color: white">.</div>

</body>
</html>
```

Listing 18: about.html

```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://thymeleaf.org">
<head>
    <meta charset="UTF-8">
    <title>All Projects</title>
    <link rel="stylesheet" type="text/css" href="/css/style.css">
    <link rel="stylesheet" type="text/css" th:href="@{/webjars/bootstrap/css/bootstrap
    .min.css}" />
</head>
<body style="background: url(/images/blurry.jpg);
    background-size: cover;
    background-repeat: no-repeat">

<!-- header -->
<div class="header">
    <div class="inner_header">
        <div class="logo_container">
            <h1>OPEN<span>PICK</span></h1>
        </div>

        <ul class="navigation">
            <a href="/projects/"><li>Projects</li></a>
            <a th:href="@{'/users/' + ${loggedInUserId}}"><li>My Profile</li></a>
            <a href="/about"><li>About</li></a>
            <a href="/logout"><li>Logout</li></a>
        </ul>

    </div>
</div>

<div style="color: white">.</div>


<div class="container-fluid text-center">
    <div><h2 style="color: white">Find your next project</h2></div>

    <div>
        <h3 class="text-success">
            [[${message}]]
        </h3>
    </div>

    <div class = "row border rounded m-3 p-3">
        <th:block th:each = "proj : ${allProjects}">
            <div class = "col">
```

```
                <div>
                    <img th:src="@{/images/project_icon.png}" width="50"/>
                </div>
                <div>
                    <a th:href="@{'/users/'+ ${proj.owner.id} + '/projects/' + ${proj.id}} "style="color:white">[[${proj.title}]]</a>
                </div>
            </div>
        </th:block>
    </div>

</div>
</body>
</html>
```

Listing 19: all$_projects.html$

```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://thymeleaf.org">
<head>
    <meta charset="UTF-8">
    <title>All Projects</title>
    <link rel="stylesheet" type="text/css" href="/css/style.css">
    <link rel="stylesheet" type="text/css" th:href="@{/webjars/bootstrap/css/bootstrap.min.css}" />
</head>
<body style="background: url(/images/blurry.jpg);
    background-size: cover;
    background-repeat: no-repeat">

<!-- header -->
<div class="header">
    <div class="inner_header">
        <div class="logo_container">
            <h1>OPEN<span>PICK</span></h1>
        </div>

        <ul class="navigation">
            <a href="/projects/"><li>Projects</li></a>
            <a th:href="@{'/users/' + ${loggedInUserId}}"><li>My Profile</li></a>
            <a href="/about"><li>About</li></a>
            <a href="/logout"><li>Logout</li></a>
        </ul>

    </div>
</div>

<div style="color: white">.</div>


<div class="container-fluid text-center">
    <div><h2 style="color: white">Find your next project</h2></div>

    <div>
        <h3 class="text-success">
            [[${message}]]
        </h3>
    </div>

    <div class = "row border rounded m-3 p-3">
        <th:block th:each = "proj : ${allProjects}">
            <div class = "col">
                <div>
                    <img th:src="@{/images/project_icon.png}" width="50"/>
                </div>
                <div>
                    <a th:href="@{'/users/'+ ${proj.owner.id} + '/projects/' + ${proj.id}} "style="color:white">[[${proj.title}]]</a>
                </div>
                <div>
                    <a th:href="@{'/users/' + ${proj.owner.id} + '/projects/' + ${proj.id} + '/delete'}" style="color:mediumpurple" >Delete</a>
```

```
                </div>
            </div>
        </th:block>
    </div>

</div>
</body>
</html>
```

Listing 20: all$_p$*rojects*$_a$*dmin.html*

```
<!DOCTYPE html>
<html lang="en">
<head>
    <link rel="stylesheet" type="text/css" href="/css/style.css">
    <title>Title</title>
</head>
<body>
    <div class="header">
        <div class="inner_header">
            <div class="logo_container">
                <h1>OPEN<span>PICK</span></h1>
            </div>

            <ul class="navigation">
                <a href="/projects"><li>Projects</li></a>
                <a href="/users/"><li>My Profile</li></a>
                <a href="/about"><li>About</li></a>
                <a href="/logout"><li>Logout</li></a>
            </ul>

        </div>

    </div>

</body>
</html>
```

Listing 21: header.html

```
<!DOCTYPE html>
<html xmlns:th="http://thymeleaf.org">
<head>
    <meta charset="UTF-8">
    <title>Open Pick</title>
    <link rel="stylesheet" type="text/css" href="/css/style.css">
    <link rel="stylesheet" type="text/css" th:href="@{/webjars/bootstrap/css/bootstrap
    .min.css}" />
</head>
<body style="background: url(/images/background.png);
    background-size: cover;
    background-repeat: no-repeat">

<!-- header -->
<div class="header">
    <div class="inner_header">
        <div class="logo_container">
            <h1>OPEN<span>PICK</span></h1>
        </div>

        <ul class="navigation">
            <a href="/about"><li>About</li></a>
        </ul>

    </div>
</div>

<div style="color: white">.</div>

<div style="text-align: center">
    <a class="h2 m-2" style="font-size: 40px; font-family: 'Calibri', sans-serif;
    background-color: white; border-radius: 5px;"th:href="@{/login}">Login</a><br>
```

```
    <a class="h2 m-2" style="font-size: 40px; font-family: 'Calibri', sans-serif;
    background-color: white; border-radius: 5px;"th:href="@{/register}">Register</a><br
    >


</div>

</body>
</html>
```

Listing 22: index.html

```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
<head>
    <meta charset="UTF-8">
    <title>Manage Users</title>
    <link rel="stylesheet" type="text/css" href="/css/style.css">
    <link rel="stylesheet" type="text/css" th:href="@{/webjars/bootstrap/css/bootstrap
    .min.css}" />
</head>
<body>

<!-- header -->
<div class="header">
    <div class="inner_header">
        <div class="logo_container">
            <h1>OPEN<span>PICK</span></h1>
        </div>

        <ul class="navigation">
            <a href="/projects/"><li>Projects</li></a>
            <a th:href="@{'/users/' + ${currentUser.id}}"><li>My Profile</li></a>
            <a href="/about"><li>About</li></a>
            <a href="/logout"><li>Logout</li></a>
        </ul>

    </div>
</div>

<div style="color: white">.</div>

<div class="container-fluid text-center">

    <!--succesful pop up -->
    <div th:if="${message}" class="alert alert-success text-center">
        [[${message}]]
    </div>

    <div>
        <h3>[[${currentUser.username}]]</h3>
    </div>

    <div>
        <h4>[[${currentUser.email}]]</h4>
    </div>

    <div>
    <a class="h4 mr-3" th:href="@{'/users/edit/' + ${currentUser.id}}">Edit</a>
    </div>

    <div class="m-2">
        <a class="h3" th:href="@{'/users/' + ${currentUser.id} + '/projects/new'}">Add
     New Project</a>
    </div>

    <div>
        <h5>Projects: </h5>
    </div>

    <div>
```

```
        <table class="table table-bordered">
            <thead class="thead-dark">
            <tr>
                <th>Title</th>
                <th>Description</th>
                <th></th>
            </tr>
            </thead>
            <tbody>
            <th:block th:each="project : ${currentUser.projects}">
                <tr>
                    <td>[[${project.title}]]</td>
                    <td>[[${project.description}]]</td>
                    <td>
                        <a class="h4 mr-3" th:href="@{'/users/' + ${currentUser.id} + '/projects/' + ${project.id}}" >Details</a>
                        <a class="h4 mr-3" th:href="@{'/users/' + ${currentUser.id} + '/projects/' + ${project.id} + '/delete'}" >Delete</a>
                        <a class="h4 mr-3" th:href="@{'/users/' + ${currentUser.id} + '/projects/' + ${project.id} + '/edit'}" >Edit</a>
                    </td>
                </tr>
            </th:block>
            </tbody>
        </table>
    </div>
</div>
</body>
</html>
```

Listing 23: profile.html

```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://thymeleaf.org">
<head>
    <meta charset="UTF-8">
    <title>New Project</title>
    <link rel="stylesheet" type="text/css" href="/css/style.css">
    <link rel="stylesheet" type="text/css" th:href="@{/webjars/bootstrap/css/bootstrap.min.css}" />
</head>
<body>

<!-- header -->
<div class="header">
    <div class="inner_header">
        <div class="logo_container">
            <h1>OPEN<span>PICK</span></h1>
        </div>

        <ul class="navigation">
            <a href="/projects/"><li>Projects</li></a>
            <a th:href="@{'/users/' + ${currentUser.id}}"><li>My Profile</li></a>
            <a href="/about"><li>About</li></a>
            <a href="/logout"><li>Logout</li></a>
        </ul>

    </div>
</div>

<div style="color: white">.</div>

<div class="container-fluid">
    <div class="text-center"><h2>Add New Project</h2></div>

    <!--<form th:action="@{/users/save}" method="post" th:object="${user}"
          style="max-width: 500px; margin: 0 auto;"> -->

    <form th:action="@{'/users/' + ${currentUser.id} + '/projects/save'}" method="post" th:object="${projectt}"
          style="max-width: 500px; margin: 0 auto;">
```

```
        <input type="hidden" th:field="*{id}">
        <input type="hidden" th:field="*{owner}" />
        <div class="border border-secondary rounded p-3">
            <!-- code for one row of input :( -->
            <div class="form-group row">
                <label class="col-sm-4 col-form-label">Title</label>
                <div class="col-sm-8">
                    <input type="text" th:field="*{title}" class="form-control"
    required minlength="1" maxlength="25"/>
                </div>
            </div>
            <!-- code for one row of input :( -->
            <div class="form-group row">
                <label class="col-sm-4 col-form-label">Description</label>
                <div class="col-sm-8">
                    <input type="text" th:field="*{description}" class="form-control"
    required minlength="1" maxlength="255" />
                </div>
            </div>
            <!-- buttons -->
            <div class="text-center">
                <button type="submit" class="btn btn-primary m-2">Save</button>
                <button type="button" class="btn btn-secondary m-2" onclick="
    cancelForm()">Cancel</button>
            </div>
        </div>
    </form>
</div>
<script type="text/javascript">
    function cancelForm() {
        window.location = "..";
    }
</script>
</body>
</html>
```

Listing 24: profile$_{f}orm.html$

```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://thymeleaf.org">
<head>
    <meta charset="UTF-8">
    <title>New Project</title>
    <link rel="stylesheet" type="text/css" href="/css/style.css">
    <link rel="stylesheet" type="text/css" th:href="@{/webjars/bootstrap/css/bootstrap
    .min.css}" />
</head>
<body>

<!-- header -->
<div class="header">
    <div class="inner_header">
        <div class="logo_container">
            <h1>OPEN<span>PICK</span></h1>
        </div>

        <ul class="navigation">
            <a href="/projects/"><li>Projects</li></a>
            <a th:href="@{'/users/' + ${userId}}"><li>My Profile</li></a>
            <a href="/about"><li>About</li></a>
            <a href="/logout"><li>Logout</li></a>
        </ul>

    </div>
</div>

<div style="color: white">.</div>

<div class="container-fluid">
    <div class="text-center"><h2>Add New Project</h2></div>

```

```
    <!--<form th:action="@{/users/save}" method="post" th:object="${user}"
          style="max-width: 500px; margin: 0 auto;"> -->

    <form th:action="@{'/users/' + ${userId} + '/projects/' + ${projectId} + '/save'}"
     method="post" th:object="${ourProject}"
          style="max-width: 500px; margin: 0 auto;">
        <input type="hidden" th:field="*{id}">
        <input type="hidden" th:field="*{owner}" />
        <div class="border border-secondary rounded p-3">
            <!-- code for one row of input :( -->
            <div class="form-group row">
                <label class="col-sm-4 col-form-label">Title</label>
                <div class="col-sm-8">
                    <input type="text" th:field="*{title}" class="form-control"
    required minlength="1" maxlength="25"/>
                </div>
            </div>
            <!-- code for one row of input :( -->
            <div class="form-group row">
                <label class="col-sm-4 col-form-label">Description</label>
                <div class="col-sm-8">
                    <input type="text" th:field="*{description}" class="form-control"
    required minlength="1" maxlength="255" />
                </div>
            </div>
            <!-- buttons -->
            <div class="text-center">
                <button type="submit" class="btn btn-primary m-2">Save</button>
                <button type="button" class="btn btn-secondary m-2" onclick="
    cancelForm()">Cancel</button>
            </div>
        </div>
    </form>
</div>
<script type="text/javascript">
    function cancelForm() {
        window.location = "../../..";
    }
</script>
</body>
</html>
```

Listing 25: profile$_{f}orm_{e}dit.html$

```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
<head>
    <meta charset="UTF-8">
    <title>Project</title>
    <link rel="stylesheet" type="text/css" href="/css/style.css">
    <link rel="stylesheet" type="text/css" th:href="@{/webjars/bootstrap/css/bootstrap
    .min.css}" />
</head>
<body>

<!-- header -->
<div class="header">
    <div class="inner_header">
        <div class="logo_container">
            <h1>OPEN<span>PICK</span></h1>
        </div>

        <ul class="navigation">
            <a href="/projects/"><li>Projects</li></a>
            <a th:href="@{'/users/' + ${loggedInUserId}}"><li>My Profile</li></a>
            <a href="/about"><li>About</li></a>
            <a href="/logout"><li>Logout</li></a>
        </ul>

    </div>
</div>
```

```
<div style="color: white">.</div>

<div class="container-fluid text-center">

    <div>
        <h3>[[${currentProject.title}]]</h3>
    </div>

    <div>
        <h4>[[${currentProject.description}]]</h4>
    </div>

    <div>
        <h5>Tasks: </h5>
    </div>

    <div>
        <table class="table table-bordered">
            <thead class="thead-dark">
            <tr>
                <th>Title</th>
                <th>Description</th>
                <th>Start Date</th>
                <th>Deadline</th>
                <th></th>
            </tr>
            </thead>
            <tbody>
            <th:block th:each="task : ${currentProject.tasks}">
                <tr>
                    <td>[[${task.title}]]</td>
                    <td>[[${task.description}]]</td>
                    <td>[[${task.startDate}]]</td>
                    <td>[[${task.deadline}]]</td>
                </tr>
            </th:block>
            </tbody>
        </table>
    </div>

</div>

<div class = "row border rounded m-3 p-3">
    <th:block th:each = "doc : ${currentProject.documents}">
        <div class = "col">
            <div>
                <img th:src="@{/images/document.png}" width="50"/>
            </div>
            <div>
                <a th:href="@{'/download?id='+ ${doc.id}} "style="color:purple">[[${
    doc.title}]]</a>
            </div>
            <div>
                ([[${#numbers.formatInteger(doc.size,4,'COMMA')}]] bytes)
            </div>
            <div>
                <a th:href="@{'/edit/'  + ${doc.id} }" style="color:rebeccapurple" >
    edit</a>
            </div>
        </div>
    </th:block>
</div>

<form th:action = "@{'/upload/' + ${currentProject.id}}" method = "post"
      enctype="multipart/form-data"
      style="max-width: 600px; margin: 0 auto"
>
    <div class="border rounded m-3">
        <h2>Upload Your files:</h2>
        <p>
```

```
            <input type = "file" name = "document" required />
        </p>
        <p>
            <input type="submit" value="Upload" class="btn btn-primary">
        </p>
    </div>

</form>

</body>
</html>
```

Listing 26: profile$_{v}iew_{o}ther.html$

```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
<head>
    <meta charset="UTF-8">
    <title>Project</title>
    <link rel="stylesheet" type="text/css" href="/css/style.css">
    <link rel="stylesheet" type="text/css" th:href="@{/webjars/bootstrap/css/bootstrap
    .min.css}" />
</head>
<body>

<!-- header -->
<div class="header">
    <div class="inner_header">
        <div class="logo_container">
            <h1>OPEN<span>PICK</span></h1>
        </div>

        <ul class="navigation">
            <a href="/projects/"><li>Projects</li></a>
            <a th:href="@{'/users/' + ${loggedInUserId}}"><li>My Profile</li></a>
            <a href="/about"><li>About</li></a>
            <a href="/logout"><li>Logout</li></a>
        </ul>

    </div>
</div>

<div style="color: white">.</div>

<div class="container-fluid text-center">

    <div>
        <h3>[[${currentProject.title}]]</h3>
    </div>

    <div>
        <h4>[[${currentProject.description}]]</h4>
    </div>

    <div class="m-2">
        <a class="h3" th:href="@{'/users/' + ${currentUser.id} + '/projects/'  + ${
    currentProject.id} + '/newtask'}">Add New Task</a>
    </div>

    <div>
        <h5>Tasks: </h5>
    </div>

    <div>
        <table class="table table-bordered">
            <thead class="thead-dark">
            <tr>
                <th>Title</th>
                <th>Description</th>
                <th>Start Date</th>
                <th>Deadline</th>
```

```
                <th></th>
            </tr>
            </thead>
            <tbody>
            <th:block th:each="task : ${currentProject.tasks}">
                <tr>
                    <td>[[${task.title}]]</td>
                    <td>[[${task.description}]]</td>
                    <td>[[${task.startDate}]]</td>
                    <td>[[${task.deadline}]]</td>
                    <td>
                        <a clas="h4 mr-3" th:href="@{'/users/' + ${currentUser.id} + '/projects/' + ${currentProject.id} + '/deletetask/' + ${task.id}}" >Delete</a>
                    </td>
                    <td>
                        <a clas="h4 mr-3" th:href="@{'/users/' + ${currentUser.id} + '/projects/' + ${currentProject.id} + '/edittask/' + ${task.id}}" >Edit</a>
                    </td>
                </tr>
            </th:block>
            </tbody>
        </table>
    </div>

</div>

<div class = "row border rounded m-3 p-3">
    <th:block th:each = "doc : ${currentProject.documents}">
        <div class = "col">
            <div>
                <img th:src="@{/images/document.png}" width="50"/>
            </div>
            <div>
                <a th:href="@{'/download?id='+ ${doc.id}} "style="color:purple">[[${doc.title}]]</a>
            </div>
            <div>
                ([[${#numbers.formatInteger(doc.size,4,'COMMA')}]] bytes)
            </div>
            <div>
                <a th:href="@{'/edit/'  + ${doc.id} }" style="color:rebeccapurple" >edit</a>
            </div>
            <div>
                <a th:href="@{'/delete/'  + ${doc.id}} " style="color:mediumpurple" >delete</a>
            </div>


        </div>
    </th:block>
</div>

<form th:action = "@{'/upload/' + ${currentProject.id}}" method = "post"
      enctype="multipart/form-data"
      style="max-width: 600px; margin: 0 auto"
>
    <div class="border rounded m-3">
        <h2>Upload Your files:</h2>
        <p>
            <input type = "file" name = "document" required />
        </p>
        <p>
            <input type="submit" value="Upload" class="btn btn-primary">
        </p>
    </div>

</form>

</body>
```

```
</html>
```

Listing 27: profile$_{v}iew_{o}wner.html$

```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://thymeleaf.org">
<head>
    <meta charset="UTF-8">
    <title>Add New Task</title>
    <link rel="stylesheet" type="text/css" href="/css/style.css">
    <link rel="stylesheet" type="text/css" th:href="@{/webjars/bootstrap/css/bootstrap
    .min.css}" />
</head>
<body>

<!-- header -->
<div class="header">
    <div class="inner_header">
        <div class="logo_container">
            <h1>OPEN<span>PICK</span></h1>
        </div>

        <ul class="navigation">
            <a href="/projects/"><li>Projects</li></a>
            <a th:href="@{'/users/' + ${currentUser.id}}"><li>My Profile</li></a>
            <a href="/about"><li>About</li></a>
            <a href="/logout"><li>Logout</li></a>
        </ul>

    </div>
</div>

<div style="color: white">.</div>

<div class="container-fluid">
    <div class="text-center"><h2>Add New Task</h2></div>


    <form th:action="@{'/users/' + ${currentUser.id} + '/projects/' + ${currentProject
    .id} + '/savetask'}" method="post" th:object="${task}"
          style="max-width: 500px; margin: 0 auto;">
        <input type="hidden" th:field="*{id}">
        <div class="border border-secondary rounded p-3">
            <!-- code for one row of input :( -->
            <div class="form-group row">
                <label class="col-sm-4 col-form-label">Title</label>
                <div class="col-sm-8">
                    <input type="text" th:field="*{title}" class="form-control"
    required minlength="1" maxlength="25"/>
                </div>
            </div>
            <!-- code for one row of input :( -->
            <div class="form-group row">
                <label class="col-sm-4 col-form-label">Description</label>
                <div class="col-sm-8">
                    <input type="text" th:field="*{description}" class="form-control"
    required minlength="1" maxlength="255" />
                </div>
            </div>
            <!-- code for one row of input :( -->
            <div class="form-group row">
                <label class="col-sm-4 col-form-label">Start Date</label>
                <div class="col-sm-8">
                    <input type="date" th:field="*{startDate}" />
                </div>
            </div>
            <!-- code for one row of input :( -->
            <div class="form-group row">
                <label class="col-sm-4 col-form-label">Deadline</label>
                <div class="col-sm-8">
                    <input type="date" th:field="*{deadline}"/>
```

```
                </div>
            </div>
            <!-- buttons -->
            <div class="text-center">
                <button type="submit" class="btn btn-primary m-2">Save</button>
                <button type="button" class="btn btn-secondary m-2" onclick="
    cancelForm()">Cancel</button>
            </div>
        </div>
    </form>
</div>
<script type="text/javascript">
    function cancelForm() {
        window.location = "..";
    }
</script>
</body>
</html>
```

Listing 28: task*form.html*

```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://thymeleaf.org">
<head>
    <meta charset="UTF-8">
    <title>Edit Task</title>
    <link rel="stylesheet" type="text/css" href="/css/style.css">
    <link rel="stylesheet" type="text/css" th:href="@{/webjars/bootstrap/css/bootstrap
    .min.css}" />
</head>
<body>
<div class="container-fluid">

    <!-- header -->
    <div class="header">
        <div class="inner_header">
            <div class="logo_container">
                <h1>OPEN<span>PICK</span></h1>
            </div>

            <ul class="navigation">
                <a href="/projects/"><li>Projects</li></a>
                <a th:href="@{'/users/' + ${currentUser.id}}"><li>My Profile</li></a>
                <a href="/about"><li>About</li></a>
                <a href="/logout"><li>Logout</li></a>
            </ul>

        </div>
    </div>

    <div style="color: white">.</div>

    <div class="text-center"><h2>Edit Task</h2></div>


    <form th:action="@{'/users/' + ${currentUser.id} + '/projects/' + ${currentProject
    .id} + '/savetask/' + ${task.id}}" method="post" th:object="${task}"
          style="max-width: 500px; margin: 0 auto;">
        <div class="border border-secondary rounded p-3">
            <!-- code for one row of input :( -->
            <div class="form-group row">
                <label class="col-sm-4 col-form-label">Title</label>
                <div class="col-sm-8">
                    <input type="text" th:field="*{title}" class="form-control"
    required minlength="1" maxlength="25"/>
                </div>
            </div>
            <!-- code for one row of input :( -->
            <div class="form-group row">
                <label class="col-sm-4 col-form-label">Description</label>
                <div class="col-sm-8">
```

```
                    <input type="text" th:field="*{description}" class="form-control"
    required minlength="1" maxlength="255" />
                </div>
            </div>
            <!-- code for one row of input :( -->
            <div class="form-group row">
                <label class="col-sm-4 col-form-label">Start Date</label>
                <div class="col-sm-8">
                    <input type="date" th:field="*{startDate}" />
                </div>
            </div>
            <!-- code for one row of input :( -->
            <div class="form-group row">
                <label class="col-sm-4 col-form-label">Deadline</label>
                <div class="col-sm-8">
                    <input type="date" th:field="*{deadline}"/>
                </div>
            </div>
            <!-- buttons -->
            <div class="text-center">
                <button type="submit" class="btn btn-primary m-2">Save</button>
                <button type="button" class="btn btn-secondary m-2" onclick="
    cancelForm()">Cancel</button>
            </div>
        </div>
    </form>
</div>
<script type="text/javascript">
    function cancelForm() {
        window.location = "..";
    }
</script>
</body>
</html>
```

Listing 29: task$_{f}orm_{e}dit.html$

```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://thymeleaf.org">
<head>
    <meta charset="UTF-8">
    <title>[[${pageTitle}]]</title>
    <link rel="stylesheet" type="text/css" href="/css/style.css">
    <link rel="stylesheet" type="text/css" th:href="@{/webjars/bootstrap/css/bootstrap
    .min.css}" />
</head>
<body>

<!-- header -->
<div class="header">
    <div class="inner_header">
        <div class="logo_container">
            <h1>OPEN<span>PICK</span></h1>
        </div>

        <ul class="navigation">
            <a href="/projects/"><li>Projects</li></a>
            <a href="/about"><li>About</li></a>
            <a href="/logout"><li>Logout</li></a>
        </ul>

    </div>
</div>

<div style="color: white">.</div>

<div class="container-fluid">

    <div class="text-center"><h2>[[${pageTitle}]]</h2></div>
    <!--<input type="hidden" th:field="*{id}"> -->
    <!--<input type="hidden" th:field="*{id}" /> -->
```

```
34
35      <!--<form th:action="@{/users/save}" method="post" th:object="${user}"
36            style="max-width: 500px; margin: 0 auto;"> -->
37
38      <form th:action="@{'/users/save/' + ${oldId}}" method="post" th:object="${user}"
39          style="max-width: 500px; margin: 0 auto;">
40          <div class="border border-secondary rounded p-3">
41              <!-- code for one row of input :( -->
42              <div class="form-group row">
43                  <label class="col-sm-4 col-form-label">E-mail</label>
44                  <div class="col-sm-8">
45                      <input type="email" th:field="*{email}" class="form-control"
    required minlength="6" maxlength="45"/>
46                  </div>
47              </div>
48              <!-- code for one row of input :( -->
49              <div class="form-group row">
50                  <label class="col-sm-4 col-form-label">Username</label>
51                  <div class="col-sm-8">
52                      <input type="text" th:field="*{username}" class="form-control"
    required minlength="1" maxlength="45" />
53                  </div>
54              </div>
55              <!-- code for one row of input :( -->
56              <div class="form-group row">
57                  <label class="col-sm-4 col-form-label">Password</label>
58                  <div class="col-sm-8">
59                      <input type="password" th:field="*{password}" class="form-control"
     required minlength="8" maxlength="15"/>
60                  </div>
61              </div>
62              <!-- code for one row of input :( -->
63              <div class="form-group row">
64                  <label class="col-sm-4 col-form-label">Enabled</label>
65                  <div class="col-sm-8">
66                      <input type="checkbox" th:field="*{enabled}" class="form-control"
    />
67                  </div>
68              </div>
69              <!-- buttons -->
70              <div class="text-center">
71                  <button type="submit" class="btn btn-primary m-2">Save</button>
72                  <button type="button" class="btn btn-secondary m-2" onclick="
    cancelForm()">Cancel</button>
73              </div>
74          </div>
75      </form>
76  </div>
77  <script type="text/javascript">
78      function cancelForm() {
79          window.location = "[[@{/users}]]";
80      }
81  </script>
82  </body>
83  </html>
```

Listing 30: user$_f orm.html$

```
1   <!DOCTYPE html>
2   <html lang="en" xmlns:th="http://thymeleaf.org">
3   <head>
4       <meta charset="UTF-8">
5       <title>Manage Users</title>
6       <link rel="stylesheet" type="text/css" href="/css/style.css">
7       <link rel="stylesheet" type="text/css" th:href="@{/webjars/bootstrap/css/bootstrap
        .min.css}" />
8   </head>
9   <body>
10
11  <!-- header -->
12  <div class="header">
```

```
13      <div class="inner_header">
14          <div class="logo_container">
15              <h1>OPEN<span>PICK</span></h1>
16          </div>
17
18          <ul class="navigation">
19              <a href="/projects/"><li>Projects</li></a>
20              <a href="/users"><li>Users</li></a>
21              <a href="/about"><li>About</li></a>
22              <a href="/logout"><li>Logout</li></a>
23          </ul>
24
25      </div>
26  </div>
27
28  <div style="color: white">.</div>
29
30  <div class="container-fluid text-center">
31      <div><h2>Manage Users</h2></div>
32
33      <!--succesful pop up -->
34      <div th:if="${message}" class="alert alert-success text-center">
35          [[${message}]]
36      </div>
37
38      <div>
39          <table class="table table-bordered">
40              <thead class="thead-dark">
41              <tr>
42                  <th>ID</th>
43                  <th>E-mail</th>
44                  <th>Username</th>
45                  <th>Enabled</th>
46                  <th></th>
47              </tr>
48              </thead>
49              <tbody>
50              <th:block th:each="user : ${listUsers}">
51              <tr>
52                  <td>[[${user.id}]]</td>
53                  <td>[[${user.email}]]</td>
54                  <td>[[${user.username}]]</td>
55                  <td>[[${user.enabled}]]</td>
56                  <td>
57  <!--                    <a class="h4 mr-3" th:href="@{'/users/' + ${user.id}}">Profile
        </a>-->
58  <!--                    <a class="h4 mr-3" th:href="@{'/users/edit/' + ${user.id}}">
        Edit</a>-->
59                      <a class="h4" th:href="@{'users/delete/' + ${user.id}}">Delete</a>
60                  </td>
61              </tr>
62              </th:block>
63              </tbody>
64          </table>
65      </div>
66  </div>
67  </body>
68  </html>
```

Listing 31: users.html

References

[1] Java Spring Introduction into Web App Development: Spring Boot CRUD Tutorial with IntelliJ IDEA, MySQL, JPA, Hibernate, Thymeleaf and Bootstrap https://www.youtube.com/watch?v=u8a25mQcMOI&t=2075s

[2] Html Header Creation https://www.youtube.com/watch?v=GxwHXxumdQk

[3] OneToMany Understanding https://www.youtube.com/watch?v=8qhaDBCJh6I

[4] OneToMany Relationship Implementation https://www.youtube.com/watch?v=ctwRpskAeIU&t=6751s

[5] Spring Security Redirect Users After Login Based on Roleshttps://www.codejava.net/frameworks/spring-boot/redirect-users-after-login-based-on-roles

[6] Spring Boot File Upload and Download with Database https://www.youtube.com/watch?v=ryRQ6qXLLYM

[7] Spring Boot Registration and Login with MySQL Database https://www.codejava.net/frameworks/spring-boot/user-registration-and-login-tutorial

[8] Spring Boot Security Role-based Authorization https://www.codejava.net/frameworks/spring-boot/spring-boot-security-role-based-authorization-tutorial

[9] Spring Security Add Roles to User https://www.codejava.net/frameworks/spring-boot/spring-boot-security-r