```java

package acme.features.manager.assignment;

import java.util.Collection;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import acme.client.data.models.Dataset;
import acme.client.services.AbstractService;
import acme.client.views.SelectChoices;
import acme.entities.project.Assignment;
import acme.entities.project.Project;
import acme.entities.project.UserStory;
import acme.features.manager.project.ManagerProjectRepository;
import acme.roles.Manager;

@Service
public class ManagerAssignmentShowService extends AbstractService<Manager, Assignment> {

    @Autowired
    private ManagerProjectRepository repository;


    @Override
    public void authorise() {
        boolean status;
        int id = super.getRequest().getData("id", int.class);
        Manager manager = this.repository.findManagerProjectByAssignmentId(id);
        status = super.getRequest().getPrincipal().getActiveRoleId() == manager.getId();

        super.getResponse().setAuthorised(status);
    }

    @Override
    public void load() {

        Assignment assigment;
        int id;

        id = super.getRequest().getData("id", int.class);
        assigment = this.repository.findAssignmentById(id);

        super.getBuffer().addData(assigment);

    }

    @Override
    public void unbind(final Assignment assigment) {
        assert assigment != null;
        int id = super.getRequest().getPrincipal().getActiveRoleId();
        SelectChoices projectChoices;
        SelectChoices userStoriesChoices;
        boolean updateable = this.repository.findProjectOfAnAssignmentByAssignmentId(assigment.getId()).isDraft();

        Collection<Project> projects = this.repository.findAllProjectsByManagerId(id);
        Collection<UserStory> userStories = this.repository.findAllUserStoriesOfAManagerById(id);

        Dataset dataset;

        projectChoices = SelectChoices.from(projects, "title", assigment.getProject());
        userStoriesChoices = SelectChoices.from(userStories, "title", assigment.getUserStory());

        dataset = super.unbind(assigment, "project", "userStory");

        dataset.put("projects", projectChoices);
        dataset.put("userStories", userStoriesChoices);
        dataset.put("updateable", updateable);

        super.getResponse().addData(dataset);

    }

}

```