```java

package acme.features.manager.userStory;

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.UserStory;
import acme.entities.project.UserStory.Priority;
import acme.roles.Manager;

@Service
public class ManagerUserStoryDeleteService extends AbstractService<Manager, UserStory> {

    // Internal state ---------------------------------------------------------

    @Autowired
    private ManagerUserStoryRepository uSRepository;

    // AbstractService<Manager, Project> --------------------------------------


    @Override
    public void authorise() {
        boolean status;
        int masterId;
        UserStory userStory;

        masterId = super.getRequest().getData("id", int.class);
        userStory = this.uSRepository.findUserStoryById(masterId);
        status = userStory != null && userStory.isDraft() && super.getRequest().getPrincipal().hasRole(Manager.class) && userStory.getManager().getId() == super.getRequest().getPrincipal().getActiveRoleId();

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

    @Override
    public void load() {
        UserStory userStory;
        int id;

        id = super.getRequest().getData("id", int.class);
        userStory = this.uSRepository.findUserStoryById(id);

        super.getBuffer().addData(userStory);
    }

    @Override
    public void bind(final UserStory userStory) {
        assert userStory != null;

        super.bind(userStory, "title", "description", "estimatedCost", "priority", "acceptanceCriteria", "link", "isDraft");
    }

    @Override
```

```
60      public void validate(final UserStory userStory) {
61          assert userStory != null;
62
63      }
64
65      @Override
66      public void perform(final UserStory userStory) {
67          assert userStory != null;
68          Collection<Assignment> assingments =
    this.uSRepository.findAllAssignmentOfAnUserStoryById(userStory.getId());
69
70          this.uSRepository.deleteAll(assingments);
71          this.uSRepository.delete(userStory);
72      }
73
74      @Override
75      public void unbind(final UserStory userStory) {
76          assert userStory != null;
77          SelectChoices priorities = SelectChoices.from(Priority.class, userStory.getPriority
    ());
78
79          Dataset dataset;
80
81          dataset = super.unbind(userStory, "title", "description", "estimatedCost",
    "priority", "acceptanceCriteria", "link", "isDraft");
82          dataset.put("priorities", priorities);
83          dataset.put("isDraft", userStory.isDraft());
84          super.getResponse().addData(dataset);
85      }
86
87  }
88
```