```java

package acme.features.manager.userStory;

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

@Service
public class ManagerUserStoryUpdateService 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
    public void validate(final UserStory userStory) {
        assert userStory != null;

    }

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

        this.uSRepository.save(userStory);
    }

    @Override
    public void unbind(final UserStory userStory) {
        assert userStory != null;
        SelectChoices priorities = SelectChoices.from(Priority.class, userStory.getPriority());

        Dataset dataset;

        dataset = super.unbind(userStory, "title", "description", "estimatedCost", "priority", "acceptanceCriteria", "link", "isDraft");
        dataset.put("priorities", priorities);
        dataset.put("isDraft", userStory.isDraft());
        super.getResponse().addData(dataset);
    }
}
```