**Описание системы**

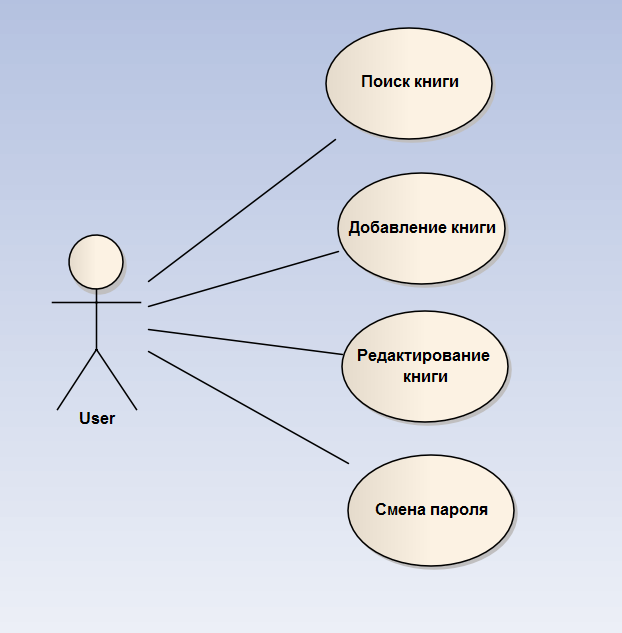
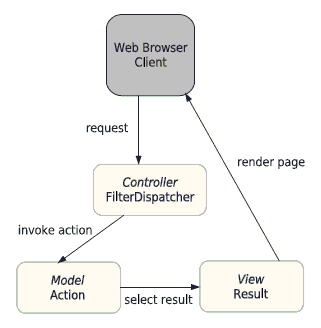
Разработанная система является web-приложением для работы с каталогом книг. Система позволяет авторизованным пользователям добалять книги в со стандартынм списоком атрибутов в каталог, а так же редактировать существующие книги и осуществлять поиск по различным атрибутам (название, автор, издательство, ISBN). Ниже представлена use case диаграмма.

Рис.1. Диаграмма вариантов использования

**Архитектура системы**

Система разработана с использованием фреймворака **Apache Struts 2**, основанного на технологии JSP и базирующегося на парадигме **Model-View-Controller**. Подробнее с архитектурой Struts 2 можно ознакомиться в [официальной документации](http://struts.apache.org/release/2.0.x/docs/architecture.html).

Рис.2. Struts 2 MVC

**Компоненты модели** представлены объектами бизнес-логики приложения (Entity Bean) и сервисными классами (Entity Access Object), которые реализую CRUD. Ниже на диаграмме представлены классы модели.

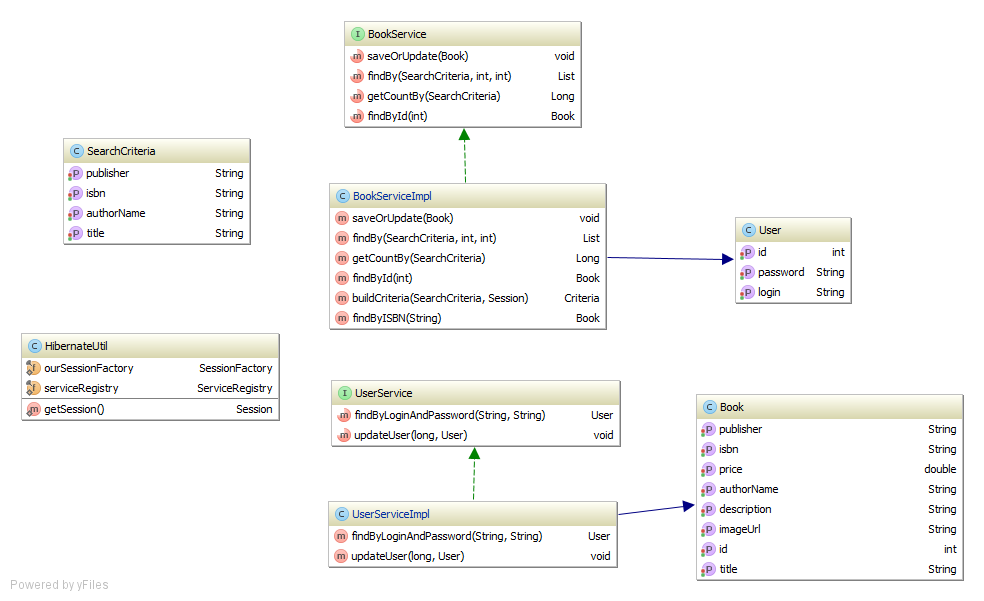


Рис.3. Классы компонеты модели

**Контроллеры** координируют деятельность приложения. В териминал Struts контроллеры включают в себя классы-action'ы и конфигурационные файлы.

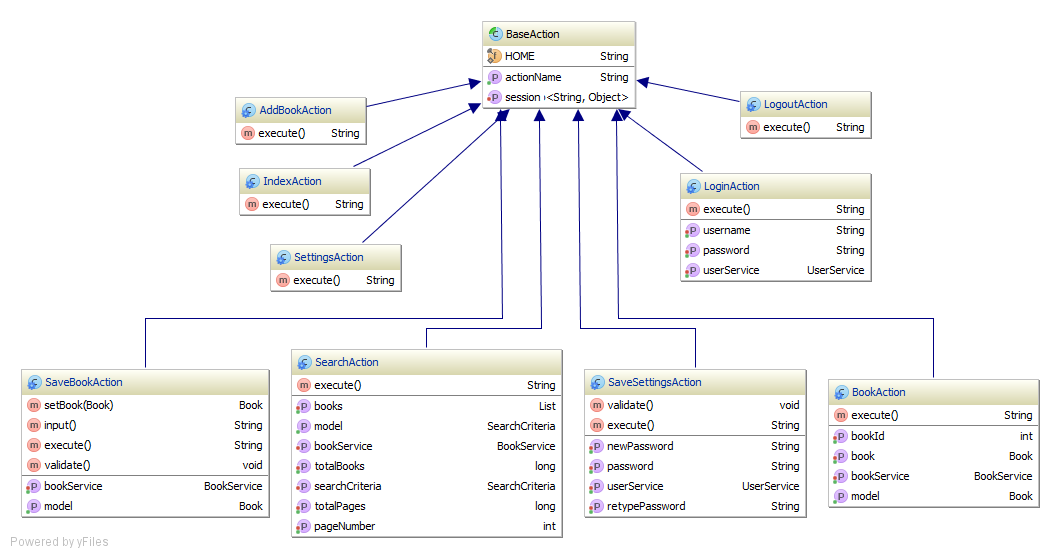
****

Рис.4. Иерархия классов Actions

**Представления** отвечают за презентацию информации и ее прием от пользователя. В Struts 2 компоненты представления соответствуют WEB-страницам, создаются при помощи JavaServer Pages (JSP) и Struts Tags, которые расширяют обычные возможности JSP и упрощают разработку компонентов представления.

**Слой базы данных.** Для хранения данных используется **RDBMS MySQL.** На верхнем уровне доступ осуществляется с помощью **ORM Hibernate.** Данныепо книгам хранятся в таблице “book”, по пользователям в таблице “user”. Типы таблиц InnoDB.

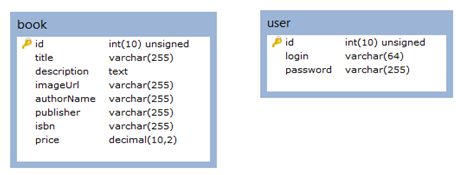


Рис.5. ER-диаграмма БД.

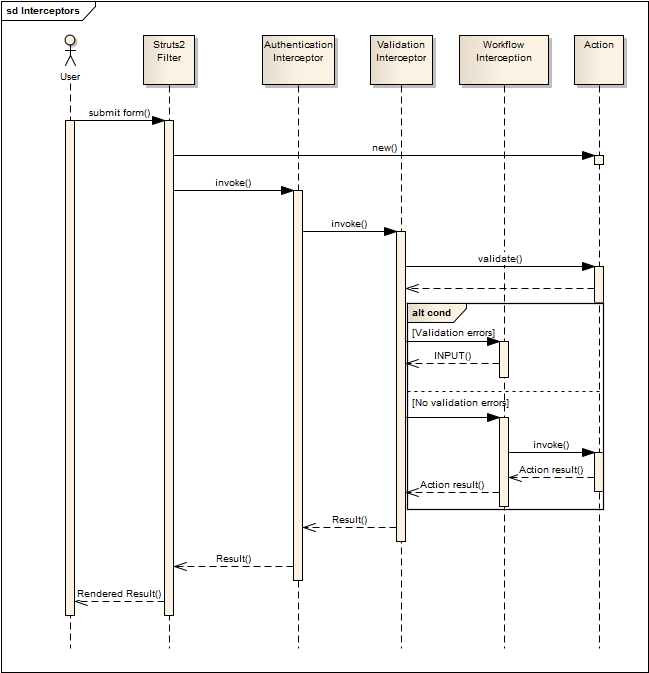
****Проверка авторизованных пользователей и валидация входных данных осуществляется с помощью механизма перехватчиков (Interceptors).

Рис.6. Диаграмма последовательности. Перехватчики.

Перехватчики – это объекты, которые вызываются до и после выполнения кода в Actions. Стек перехватчиков конфигурируется на уровне struts.xml.

На диаграмме Рис.6 представлена последователность вызовов перехватчиков с момента отправки пользователем формы до возврата результата выполнеия запроса в браузер.

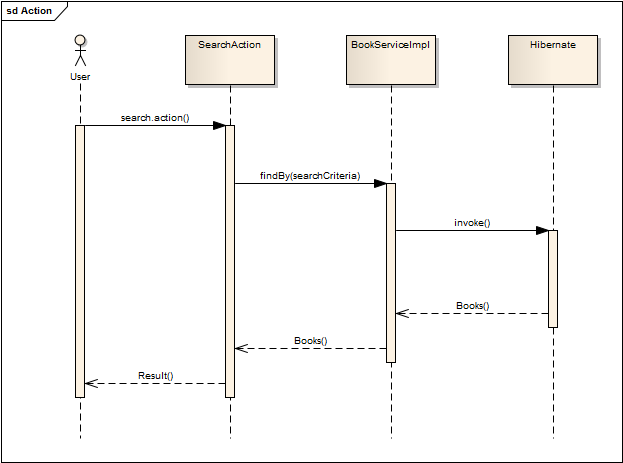
****

Рис.7. Диграмма последовательности. Поиск книг в каталоге

**Струкрты каталогов проекта.**

|  |  |
| --- | --- |
| **install** | содержит скрипты, необходимые для установки приложения |
| **lib** | содержит используемые библиотеки |
| **resources** | каталог ресурсов приложения |
| hibernate.cfg.xml | файл конфигурации Hibernate |
| struts.xml | файл конфигурации Struts |
| messages.properties | содержит строки-сообщения |
| **src** | содежит исходный код приложения |
| **com.app** |  |
| **actions** | пакет, содержащий классы Action |
| **model** | пакет, содержащий сущности приложения |
| **services** | пакет, содержащий классы сервисы |
| **util** | пакет, содержащий вспомогательные классы |
| **tests** | модульные тесты |
| **com.app.actions** | тесты для классов Action |
| **com.app.** **services** | тесты CRUD, DBUnit |
| **web** |  |
| **assets** | содержит статику (js, css) |
| **view** | jsp-файлы представлений |
| **layouts** | шаблоны представлений |
| **WEB-INF** | метаданные приложения |
| web.xml | конфигурацонный файл приложения |
| **docs** | содержит документацию |
| **diagramms** | UML диаграммы |
| **javadoc** | документация JavaDoc |
| **mockups** | мокапы пользовательского интерфейса |

**Тестирование**

**Используемые технологии и средства**

* Java SE 7, JSP
* Apache Struts 2.3.1
* Hibernate 4 ORM
* JUnit
* MySQL 5.1
* Tomcat 7
* IDEA 12.1
* Maven
* Team City
* Git

**Инструкция по развертыванию**

1. Конфигурирование
2. Сборка, подтягивание зависимостей
3. Деплой на сервер
4. Установка базы
5. Проверка, тесты
6. Логи?