

## Описание системы

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

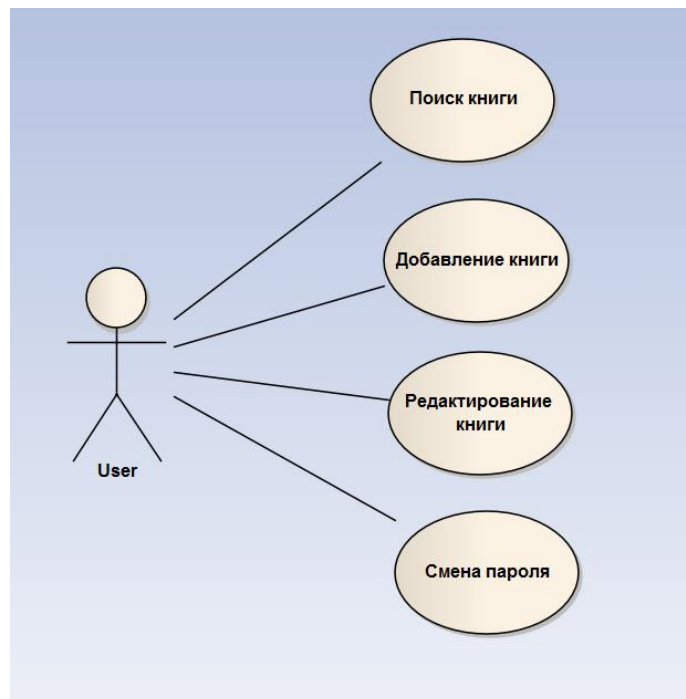


Рис.1. Диаграмма вариантов использования

## Архитектура системы

Система разработана с использованием фреймворка **Apache Struts 2**, основанного на технологии JSP и базирующегося на парадигме **Model-View-Controller**. Подробнее с архитектурой Struts 2 можно ознакомиться в [официальной документации](#).

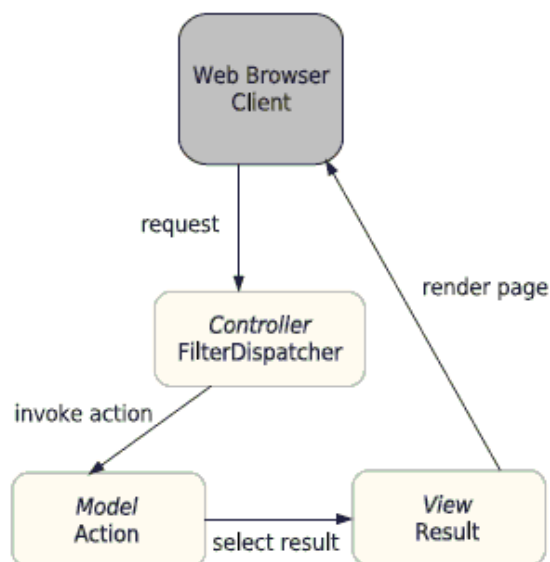


Рис.2. Struts 2 MVC

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

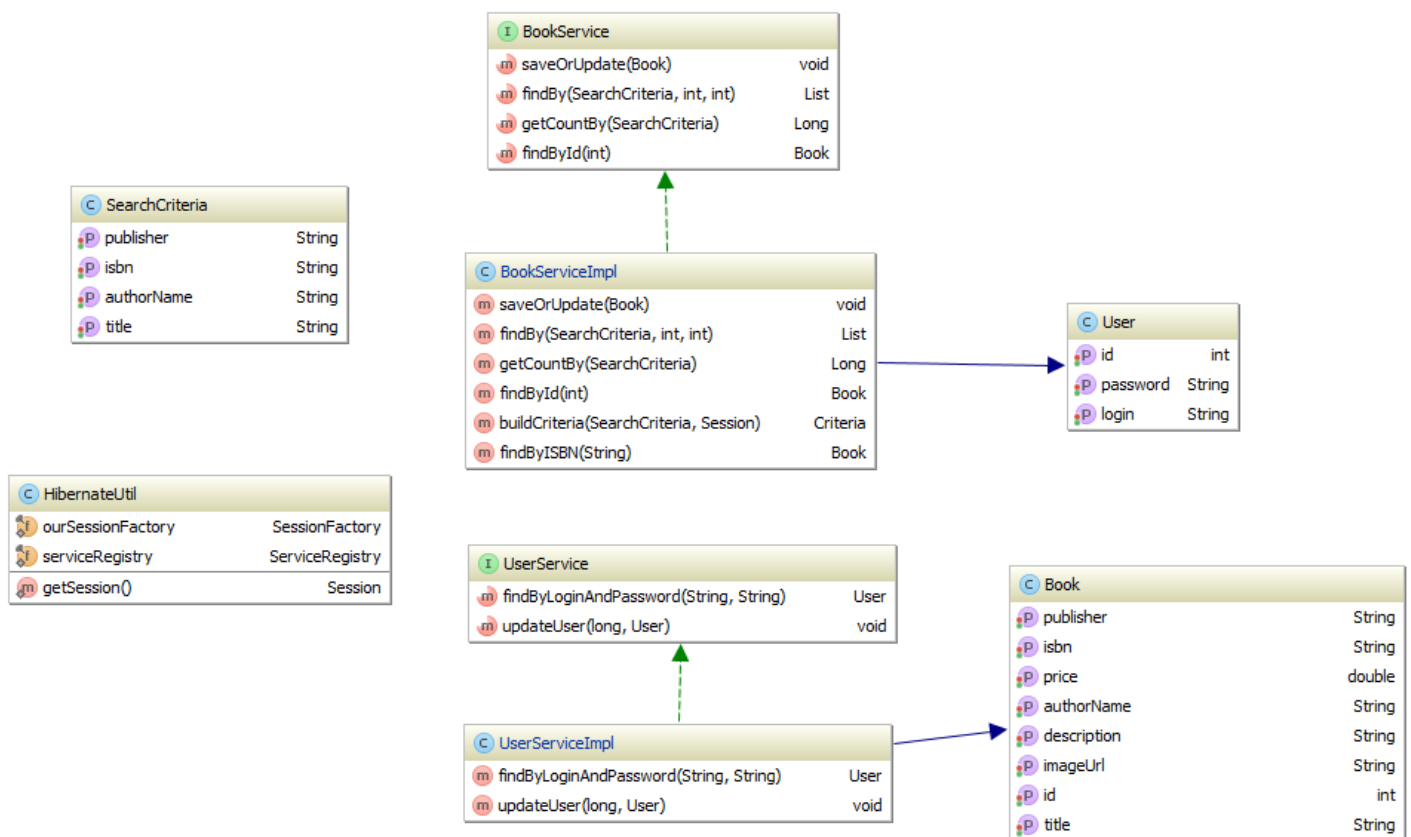


Рис.3. Классы компоненты модели

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

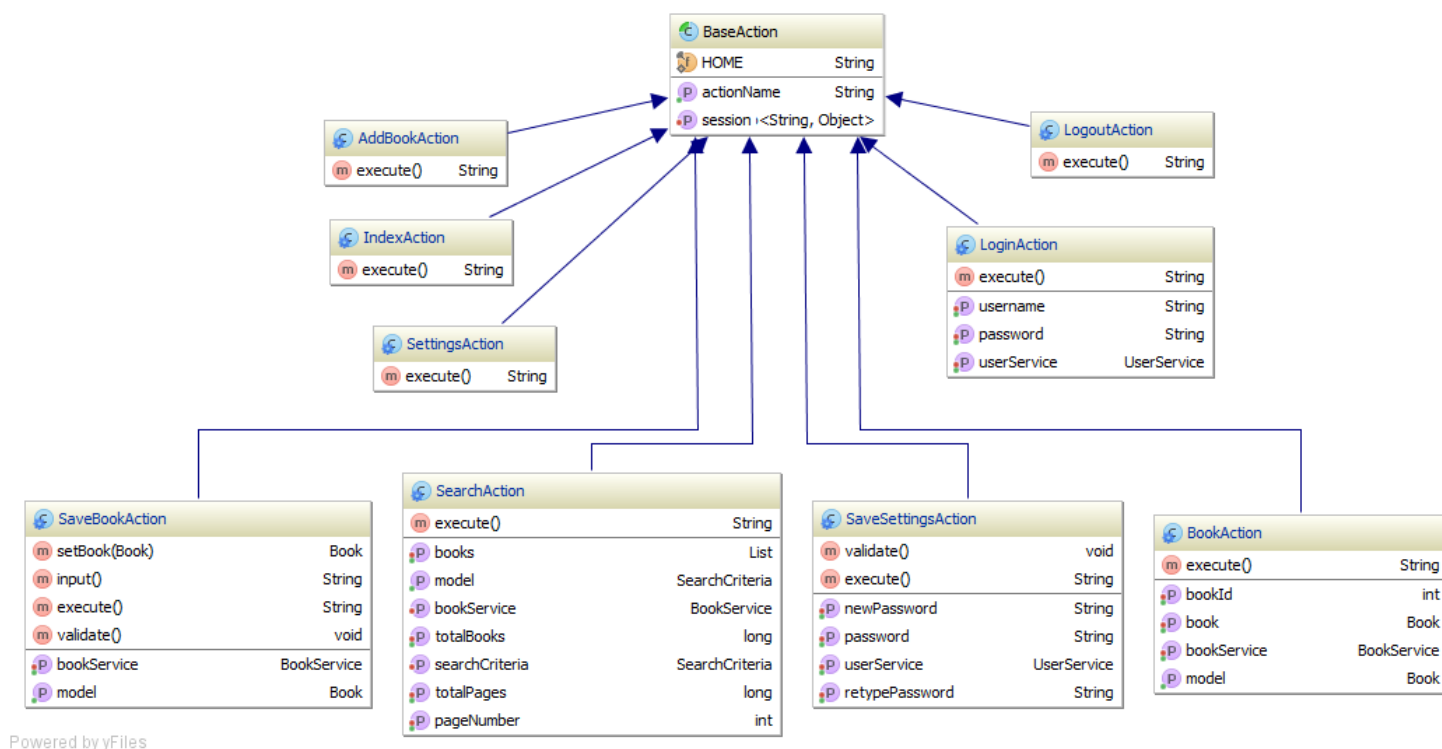


Рис.4. Иерархия классов Actions

**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**. Данные по книгам хранятся в таблице “book”, по пользователям в таблице “user”. Типы таблиц InnoDB.

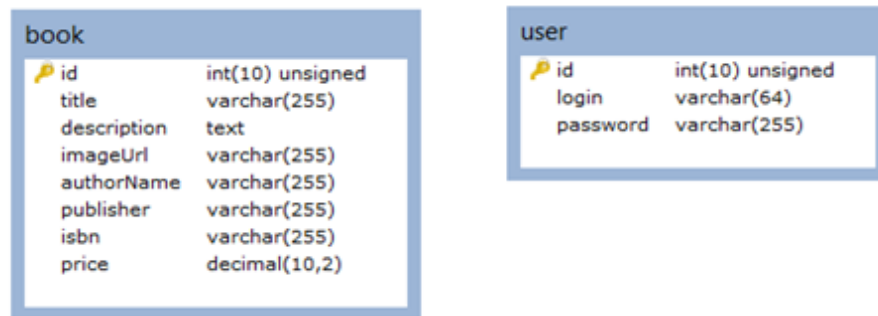


Рис.5. ER-диаграмма БД.

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

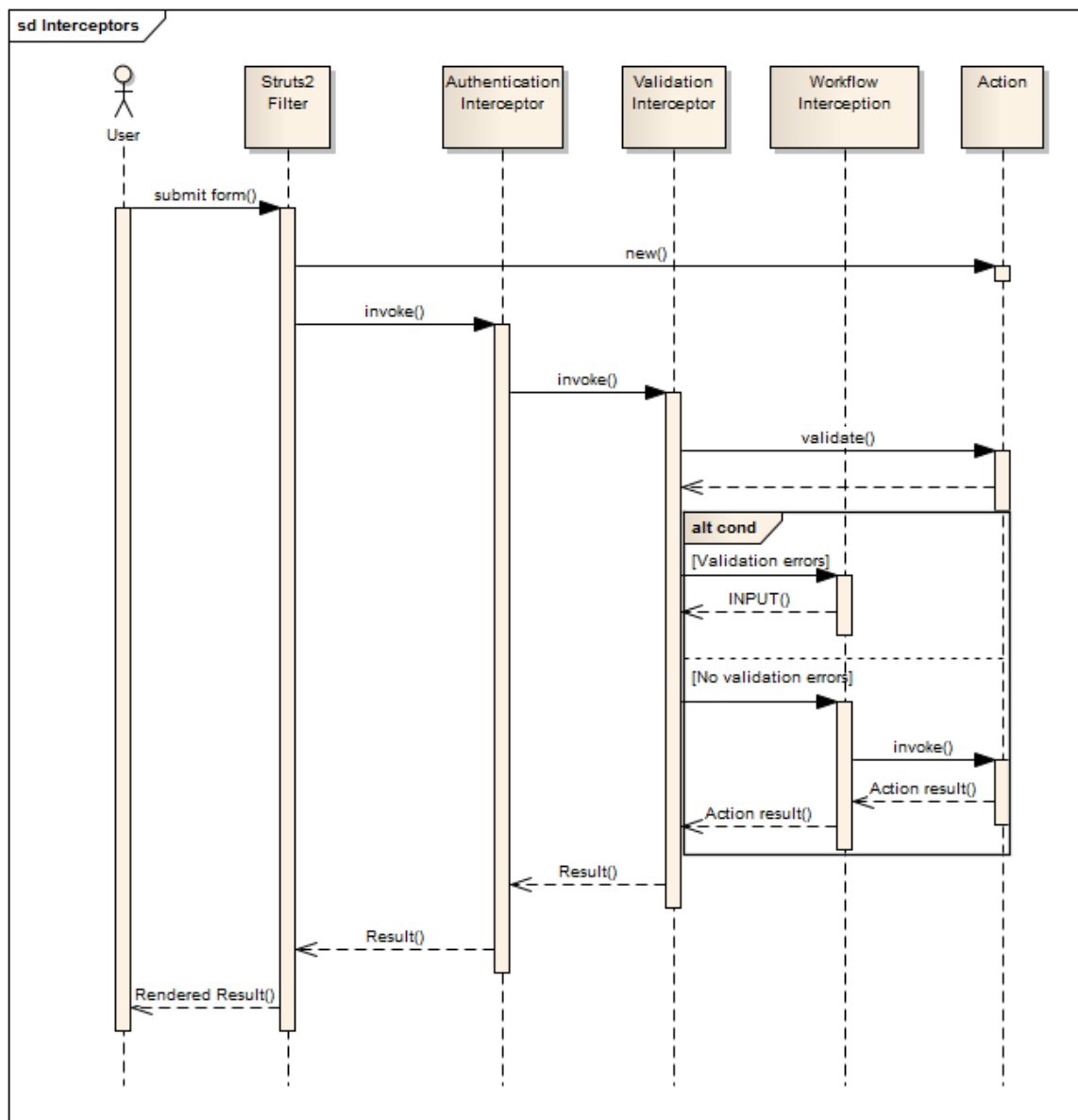


Рис.6. Диаграмма последовательности. Перехватчики.

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

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

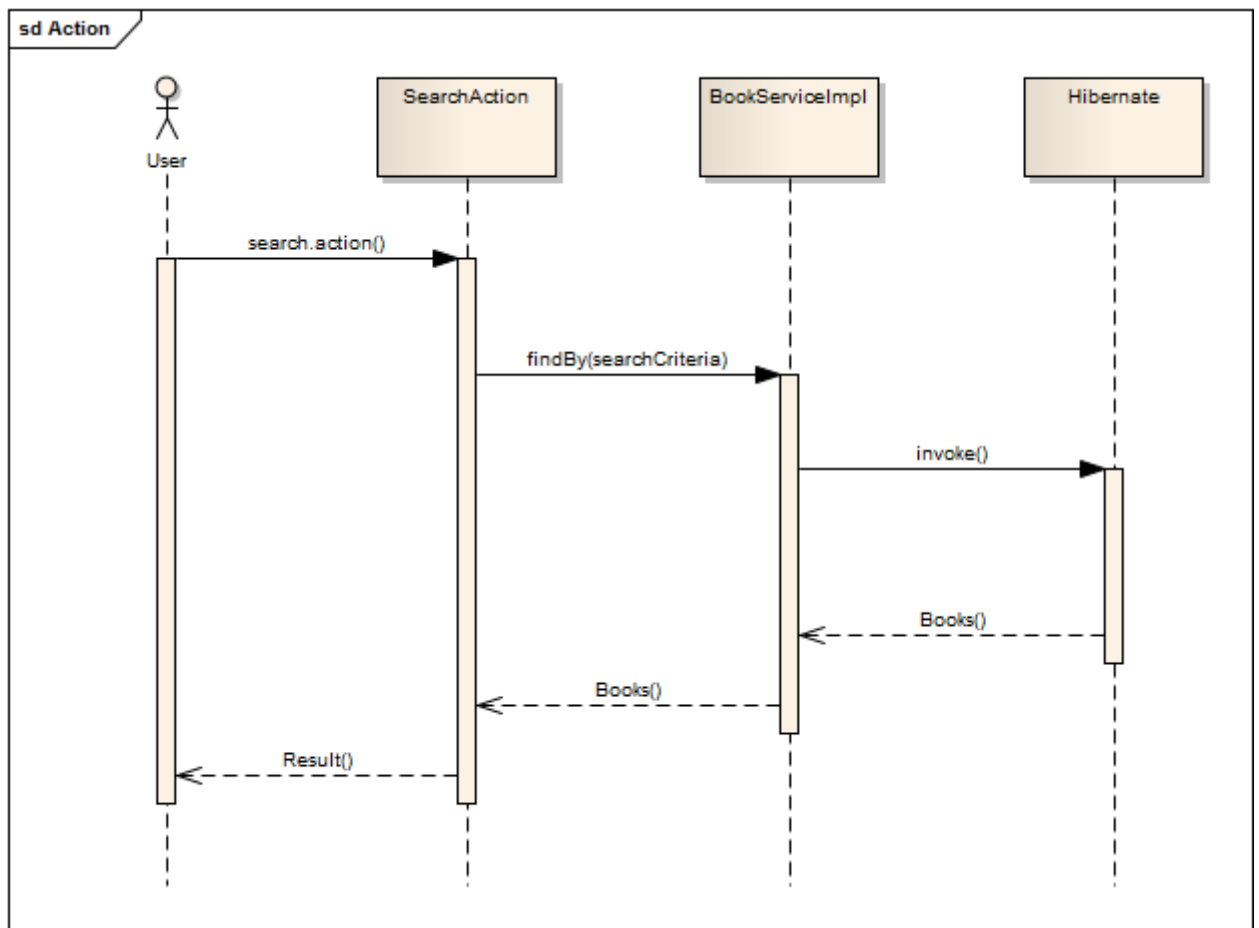


Рис.7. Диаграмма последовательности. Поиск книг в каталоге

## Структура каталогов проекта.

<b>install</b>	содержит скрипты, необходимые для установки приложения
<b>src</b>	содержит исходный код приложения
<b>main</b>	
<b>java</b>	
<b>com.app.actions</b>	пакет, содержащий классы Action
<b>com.app.model</b>	пакет, содержащий сущности приложения
<b>com.app.services</b>	пакет, содержащий классы сервисы
<b>com.app.util</b>	пакет, содержащий вспомогательные классы
<b>resources</b>	каталог ресурсов приложения
hibernate.cfg.xml	файл конфигурации Hibernate
struts.xml	файл конфигурации Struts
messages.properties	содержит строки-сообщения
<b>webapp</b>	
<b>assets</b>	содержит статику (js, css)
<b>views</b>	jsp-файлы представлений
<b>layouts</b>	шаблоны представлений
<b>WEB-INF</b>	метаданные приложения
web.xml	конфигурационный файл приложения
<b>test</b>	модульные тесты
<b>java</b>	
<b>com.app.actions</b>	тесты классов Action
<b>com.app.services</b>	тесты CRUD, DBUnit
<b>com.app.util</b>	тесты вспомогательных классов
<b>docs</b>	содержит документацию
<b>diagramms</b>	UML диаграммы
<b>javadoc</b>	документация JavaDoc
<b>mockups</b>	мокапы пользовательского интерфейса

## Инструкция по развертыванию

1. Создать и наполнить БД с помощью команды

```
mysql -u USER -pPASSWORD -h HOST < install/dump.sql
```

2. Выполнить конфигурацию Hibernate, указав настройки соединения в файле к БД hibernate.cfg.xml
3. Выполнить конфигурацию модульных тестов в файле test/com/app/services/BaseDBUnitTest.java, указав в конструкторе класса настройки соединения.
4. Выполнить тестирование и сборку проекта с помощью Apache Maven

```
mvn test / mvn package
```
5. Полученный war-архив приложения задеплоить на сервер Tomcat одним из следующих способов <http://tomcat.apache.org/tomcat-7.0-doc/deployer-howto.html>
6. Включить UTF-8 кодировку для GET параметров в Tomcat. Добавить атрибут `URIEncoding="UTF-8"` в элемент `<Connector>` конфигурационного файла server.xml
7. Проверить работу приложения — по адресу <http://YOUR-HOST/BooksCatalog/> должна быть доступна форма для входа
8. Логи приложения можно найти в файле `{TomcatDir}/logs/booksCatalog.log`  
настройки уровней логирования в файле log4j.properties

## **Используемые технологии и средства**

- Java SE 7, JSP
- Apache Struts 2.3.1
- Hibernate 4 ORM
- JUnit
- MySQL 5.1
- Tomcat 7
- Twitter Bootstrap, jQuery
- Maven
- Git
- IDEA 12.1
- Balsamiq Mockups
- Sparx Enterprise Architect

## **Дополнительная информация**

- Публичный git репозиторий проекта  
<https://github.com/eugenix/BooksCatalog>
- Задеплоеный проект <http://176.34.126.240:8080/BooksCatalog/>

[admin:password]