**Санкт-Петербургский политехнический университет Петра Великого**

Институт компьютерных наук и технологий

Кафедра «Распределенные вычисления и компьютерные сети»

**ОТЧЕТ**

**по курсовой работе**

по дисциплине «Программное обеспечение распределенных вычислительных систем»

Тема работы: «Электронная библиотека»

**Работу выполнил студент** гр. 63507/1 Архипов Т. Д.

**Преподаватель** Стручков И.В.

Санкт-Петербург

2017

Оглавление

[Введение 3](#_Toc479820396)

[Формулировка задания 3](#_Toc479820397)

[Роли в проекте 3](#_Toc479820398)

[*Система*: 3](#_Toc479820399)

[*Пользователь:* 4](#_Toc479820400)

[*Администратор:* 4](#_Toc479820401)

[Варианты использования 4](#_Toc479820402)

[*Регистрация пользователя* 4](#_Toc479820403)

[*Авторизация пользователя* 5](#_Toc479820404)

[*Поиск книги по автору и названию* 5](#_Toc479820405)

[Заказ книги(при наличии свободных экземпляров) 5](#_Toc479820406)

[*Возврат книги* 6](#_Toc479820407)

[Модель предметной области 6](#_Toc479820408)

[Диаграмма классов 7](#_Toc479820409)

[Диаграммы последовательности 7](#_Toc479820410)

[Реализация 12](#_Toc479820411)

[*Слой бизнес-логики* 12](#_Toc479820412)

[*Слой представления* 12](#_Toc479820413)

[Инструкция системного администратора 17](#_Toc479820414)

[Инструкция пользователя 17](#_Toc479820415)

[Выводы 17](#_Toc479820416)

## Введение

В рамках курса «программное обеспечение распределенных вычислительных систем» было необходимо разработать приложение, демонстрирующее понимание студентом принципов разработки программного обеспечения. Необходимо, чтобы разработанное приложение содержало три следующих слоя:

* слой бизнес-логики;
* слой хранения данных;
* слой представления.

В качестве примера задачи по курсу была выбрана задача реализации многопользовательской электронной библиотеки. Библиотека предоставляет пользователям возможность регистрации в системе, возможность поиска книг по всем библиотекам с возможностью последующего их заказа (в дальнейшем пользователю необходимо будет забрать книгу из библиотеки), а также возможность оформления возврата книги в библиотеку.

В ходе работы требовалось выполнить следующие задачи:

* Научиться разрабатывать web-приложения с помощью языка разработки Java в среде NetBeans.
* Исследовать использование технологии хранения данных (persistence) Java EE 5 для размещения данных приложения в СУБД.
* Овладеть навыками по работе с компонентами системы EJB, необходимыми для создания, развертывания и функционирования распределенных приложений Java масштаба предприятия.
* Проявить способности в обеспечении клиентского доступа к системе с помощью компоненты Java Server Pages.

## Формулировка задания

Приложение «электронная библиотека» должно обеспечивать возможность пользователям для заказа книг в библиотеке. Для простоты и удобства работы система должна предоставлять пользователям возможность регистрации в системе электронной библиотеки, возможность поиска книг, бронирования книг, присутствующих в одной из библиотек, а также возможность для оформления их возврата в библиотеку. Для удобства работы пользователя с библиотекой в системе должна поддерживаться авторизация.

Для возможности модификации данных о библиотеке, книгах и пользователях электронной библиотеки также должна поддерживаться возможность ее отдельного администрирования.

В системе необходимо реализовать следующие операции: поиск книги по названию/автору, заказ книги (при наличии свободных экземпляров), возврат книги, список книг с указанием местонахождения.

Сериализуемый объект: описание книги (автор, название, местонахождение)

## Роли в проекте

В проекте явно выделяются три роли: система, ее пользователь и ее администратор. Ниже описаны основные действия, которые должны производиться данными ролями:

### *Система*:

* Возможность регистрации пользователей
* Возможность авторизации пользователей
* Проверка корректности всех вводимых пользователем данных
* Предоставление всех книг, удовлетворяющих запросу пользователя, с указанием их местоположения
* Возможность бронирования определенной книги в определенной библиотеке
* Возможность оформления возврата книги

Возможности для администрирования данными:

* Возможность для отличия произвольных пользователей библиотеки от ее администратора
* Предоставление данных обо всех пользователях
* Возможность удаления пользователей
* Возможность добавления новой библиотеки
* Возможность просмотра библиотек
* Возможность удаления любой библиотеки
* Возможность добавления новых книг в библиотеку
* Возможность удаления определенной книги из библиотеки

### *Пользователь:*

* Регистрация нового пользователя
* Авторизация в системе электронной библиотеки
* Поиск книги
* Заказ книги
* Возврат книги

### *Администратор:*

* Добавление новой библиотеки
* Добавление новых книг
* Удаление существующей библиотеки
* Удаление существующих книг
* Удаление существующих пользователей

## Варианты использования

Ниже представлены все варианты использования системы, описывающие возможные действия пользователя системы и реакцию системы на данные действия.

Среди данного списка не представлены варианты использования системы администратором, такие как добавление и удаление библиотеки, добавление и удаление книг, удаление пользователя, а также варианты просмотра книг, заказов, библиотек или пользователей. Эти варианты не представлены ниже, так как возможность администрирования системы реализовывалась лишь дополнительно, для обеспечения полного представления возможностей использования системы рядовым пользователем.

### *Регистрация пользователя*

1. Пользователь вводит в специальное поле логин и пароль. Нажимает кнопку «зарегистрироваться».
2. Система производит проверку введенных пользователем данных:

- удовлетворяют ли введенные логин/пароль всем необходимым условиям (непустота, отсутствие спецсимволов)

- существует ли уже пользователь с данным логином

1. Система вносит данные о новом пользователе в БД и выдает подтверждение регистрации.

*Альтернатива*

3а. Если есть какие-то недочеты во введенных пользователем данных, система оповещает об этом пользователя.

### *Авторизация пользователя*

1. Пользователь вводит в специальное поле логин и пароль. Нажимает кнопку «войти».
2. Система производит проверку введенных пользователем данных.
3. Система авторизует пользователя и запоминает его имя для текущего сеанса.

*Альтернатива*

3а. Если есть какие-то недочеты во введенных пользователем данных, система оповещает об этом пользователя.

### *Поиск книги по автору и названию*

1. Пользователь вводит в специальную форму фамилию автора и название книги.
2. Пользователь выбирает параметр поиска: по точному совпадению автора/названия, либо по частичному совпадению.
3. Система производит поиск и выводит список всех библиотек, в которых есть книга с заданными параметрами.

*Альтернатива*

3а. Если не найдено ни одной книги, удовлетворяющей запросу, система выводит пустой список книг.

### Заказ книги(при наличии свободных экземпляров)

1. Пользователь вводит в специальную форму фамилию автора и название книги.
2. Система производит поиск и выводит список всех книг, удовлетворяющих запросу.
3. Пользователь выбирает интересующую его книгу и нажимает кнопку «заказать».
4. Если пользователь авторизован в системе, то производится бронирование книги на его имя.

*Альтернатива*

2а.1) Если ни одной книги, удовлетворяющей запросу, не найдено, система выводит пустой список книг

3а.1) Если пользователь не находит интересующую его книгу, он отменяет заказ.

3а.2) Система не производит бронирование.

4а.1) Если пользователь не авторизован, система запрашивает логин и пароль пользователя или предлагает регистрацию пользователя.

4а.2) Пользователь выбирает вариант авторизации и вводит логин/пароль.

4а.3) Система проверяет корректность введенных данных.

4а.4) При корректно введенных логине и пароле система производит бронирование книги в библиотеке, выдавая пользователю номер бронирования.

4б.1) Пользователь выбирает вариант регистрации.

4б.2) Открывается схема для регистрации нового пользователя.

4б.3) Система проверяет корректность введенных данных.

4б.4) При корректно введенных логине/пароле система производит бронирование книги в библиотеке, выдавая пользователю номер бронирования.

4а.4).а). Если логин/пароль введены некорректно, система не производит бронирование и оповещает об этом пользователя и предлагает ему ввести данные заново.

### *Возврат книги*

1. Если пользователь авторизован, система выводит список заказанных книг.
2. Пользователь выбирает книгу, которую хочет вернуть, и нажимает «вернуть».
3. Система записывает книгу как возвращенную.

*Альтернатива*

1а.1) Если пользователь не авторизован, система предлагает пользователю ввести номер заказа, свой логин и пароль

1а.2) Пользователь вводит номер заказа книги, которую планирует вернуть, а также свой логин и пароль.

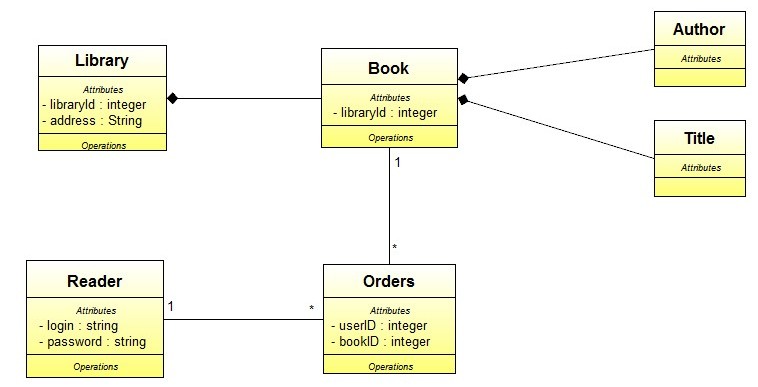
1а.3) Система производит проверку введенных пользователем данных.

1а.4) Система записывает книгу как возвращенную.

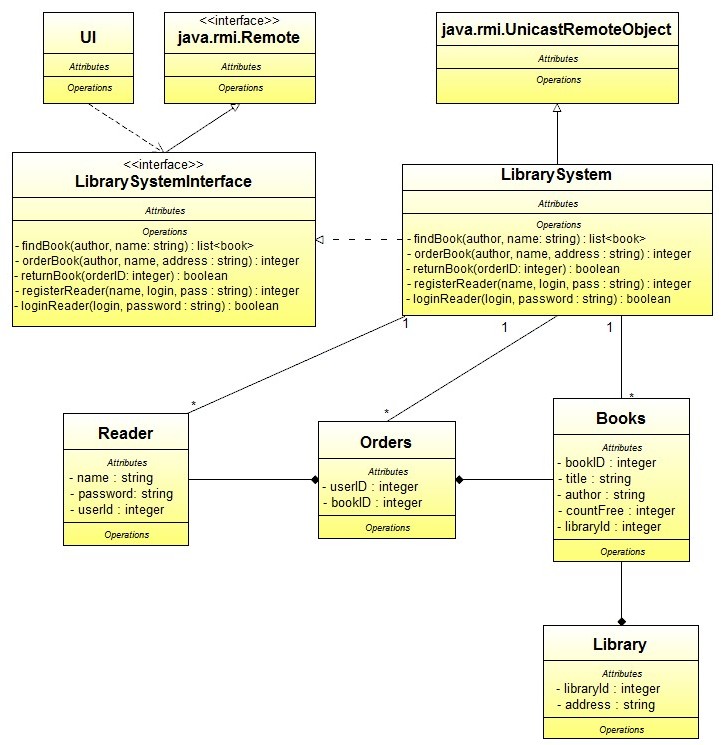
1.а.4).а). Если логин/пароль введены некорректно, система не производит бронирование и оповещает об этом пользователя и предлагает ему ввести данные заново.

## Модель предметной области

Ниже изображена модель предметной области, представляющая основные понятия предметной области и отношения между ними. В модель, помимо прочих понятий, было включено понятие заказа, связывающее конкретный экземпляр книги с некоторым читателем.



## Диаграмма классов

****

## Диаграммы последовательности

Ввиду того, что многие диаграммы последовательности не отражают важные стороны функциональности системы, ниже описаны лишь основные диаграммы последовательности. Большинство диаграмм, описывающих альтернативные сценарии, опущено из-за их неинформативности. Альтернативными в данном случае считаются сценарии, в которых пользователем вводятся некорректные данные, либо когда результатами поиска являются пустые списки.

1. Описанная ниже диаграмма последовательности предоставляет информацию о последовательности вызовов для успешного сценария регистрации нового читателя библиотеки. При этом, как видно из диаграммы, создается новый элемент класса «читатель».

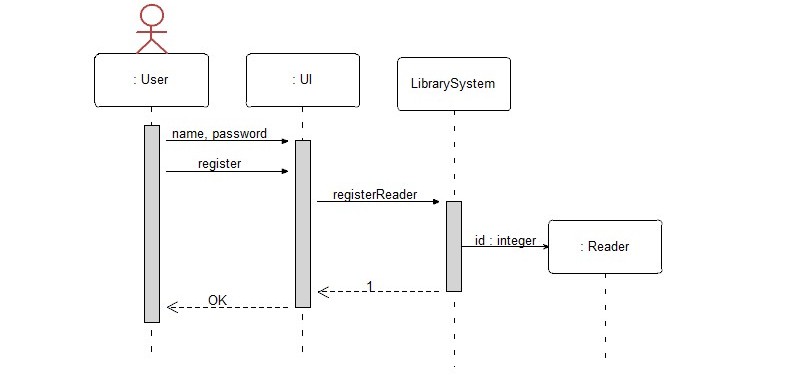
****

Рис. 3. Диаграмма последовательности: регистрация пользователя

1. Следующая диаграмма описывает последовательность взаимодействий объектов при успешном сценарии авторизации пользователя. В данном случае проверяется существование читателя с заданным именем, а также правильность введённых пользователем логина и пароля. В случае положительного результата проверки пользователю выводится сообщение об успешной авторизации, а в системе запоминается состояние с текущим именем читателя.

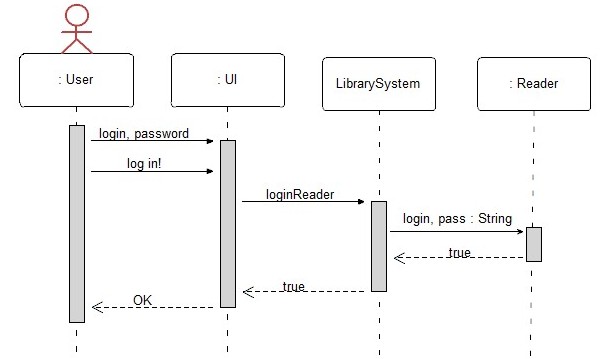
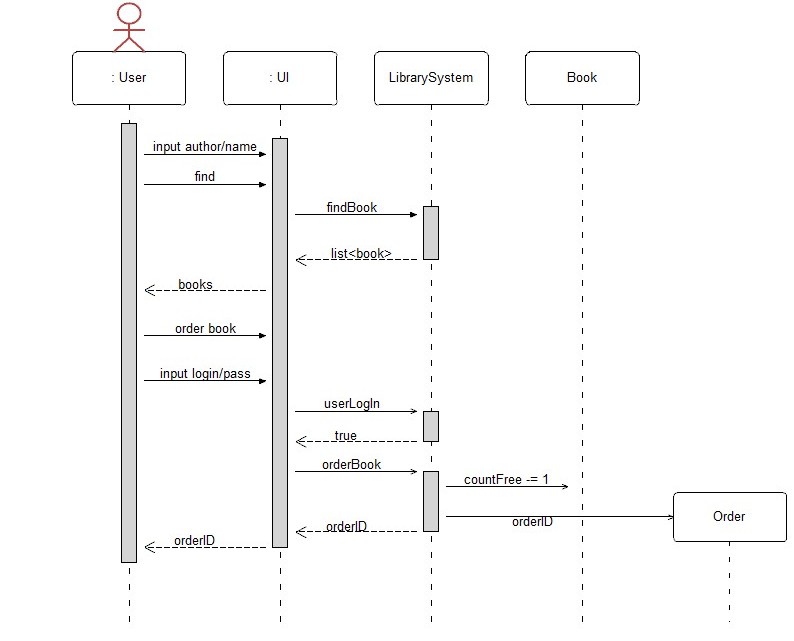


Рис. 4. Диаграмма последовательности: авторизация пользователя

1. Диаграмма на рисунке 5 описывает последовательность взаимодействия объектов при поиске книг неавторизованным читателем в системе электронной библиотеки.

Сперва пользователь вводит параметры поиска книг, после чего система выдает пользователю результаты поиска: книги с указанием адреса библиотеки, в которой эти книги находятся. Среди представленных книг пользователь выбирает книгу, которую хочет заказать, затем авторизуется в системе. После проверки введенных пользователем данных производится заказ книги на авторизовавшегося читателя, и ему сообщается присвоенный номер заказа книги.

С этим номером читатель, как и в любой аналогичной системе заказов, должен будет в будущем обратиться в библиотеку по выбранному читателем адресу.

Рис. 5. Диаграмма последовательности: заказ книги неавторизованным пользователем

1. Примерно аналогичная последовательность взаимодействий происходит при заказе книги уже авторизованным пользователем (рис. 6) и при заказе книги еще не зарегистрированным в системе пользователем с регистрацией прямо в процессе заказа книги (рис. 7).

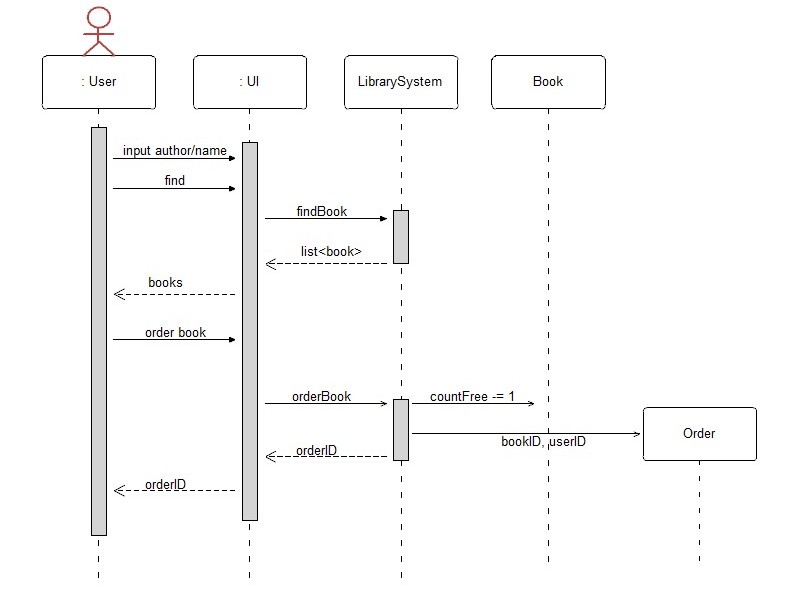
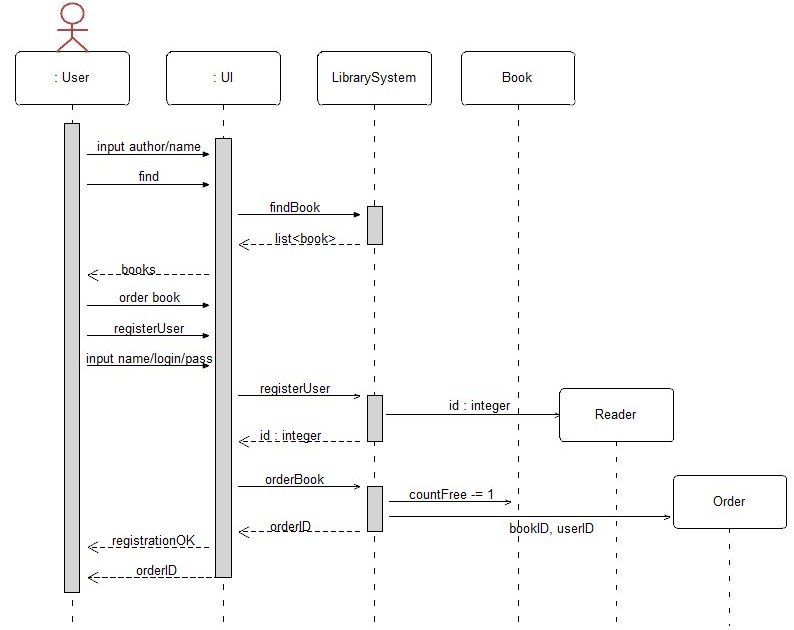
Рис. 6. Диаграмма последовательности: заказ книги авторизованным пользователем

Рис. 7. Диаграмма последовательности: заказ книги незарегистрированным пользователем

1. Диаграмма последовательности поиска книги является частью диаграммы заказа книги, однако, она должна быть представлена (рис. 8), так как поиск книг в библиотеке является отдельным примером использования системы.

Для поиска книг пользователь, не обязательно авторизованный, вводит имя автора, название книги, а также выбирает параметр поиска: по точному или частичному совпадению.

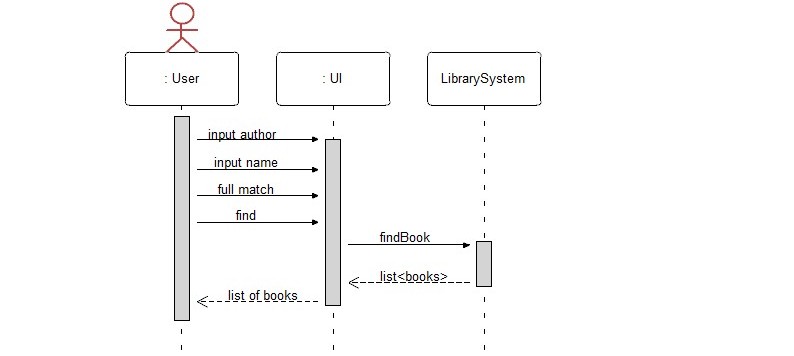


Рис. 8. Диаграмма последовательности: поиск пользователем книги

Диаграмма последовательности возврата книги описывает возврат книги авторизованным пользователем: читатель выбирает книгу, которую хочет вернуть, из списка, предложенного системой. Система генерирует список книг, опираясь на данные о заказах книг, произведенных данным читателем. При оформлении возврата система удаляет заказ из базы данных.

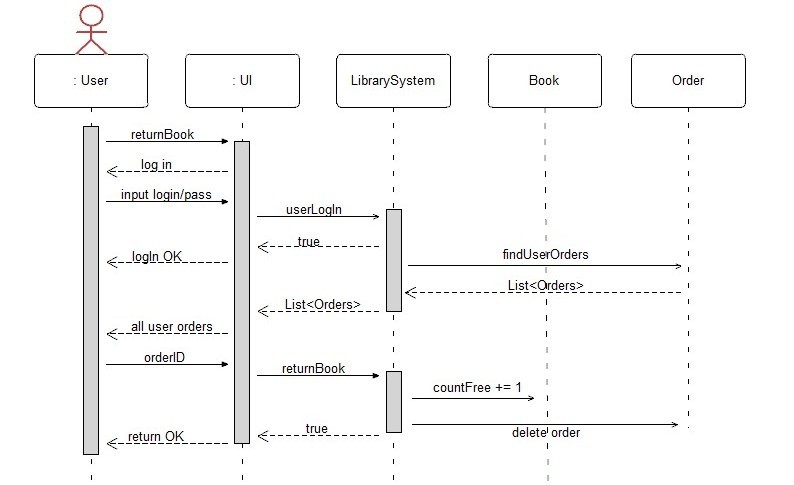


Рис. 9. Диаграмма последовательности: возврат книги неавторизованным пользователем

## Реализация

**Слой хранения данных**

Для хранения данных электронной библиотеки использовалась база данных JavaDB. При разработке в качестве основных классов в системе были взяты сущности, описанные в модели предметной области: заказ, книга, библиотека и пользователь. Для хранения данных в БД были созданы таблицы, соответствующие перечисленным выше сущностям. Ниже перечислены атрибуты каждой из таблиц:

* **BOOKS:** *bookId, author, title, libraryId, countFree*
* **LIBRARIES:** *libraryId, address*
* **LIBRARYUSER:** *userId, username, password*
* **ORDERS:** *orderId, bookId, userId*

### *Слой бизнес-логики*

Для каждой из сущностей были автоматически сгенерированы java-классы, соответствующие таблицам, созданным ранее в СУБД. Бизнес-логика расположена в пакете library\_system, содержащий следующие классы:

* Books.java – класс, соответствующий сущности «BOOKS». В данном классе формируется информация обо всех книгах, записанных в электронной библиотеке, получаемая из базы данных
* Libraryuser.java – класс, соответствующий сущности «LIBRARYUSER». В данном классе формируется информация о пользователях системы, получаемая из базы данных
* Orders.java – класс, соответствующий сущности «ORDERS». В данном классе формируется информация о заказах книг, получаемая из базы данных
* Libraries.java – класс, соответствующий сущности «LIBRARIES». В данном классе формируется информация об адресах библиотек, получаемая из базы данных.
* TotalResult.java – класс, в котором формируется итоговая статистика по игре на данный момент
* LibrarySession.java – реализация интерфейса работы с данными электронной библиотеки
* LibrarySessionRemote.java – интерфейс удаленной работы с данными электронной библиотеки
* LibrarySessionLocal.java – локальный интерфейс работы с данными электронной библиотеки
* LibraryAdministration.java - реализация интерфейса администрирования системы
* LibraryAdministrationRemote.java - интерфейс удаленного администрирования системы

### *Слой представления*

Для разработки слоя представления была выбрана технология Java Server Pages. Для реализации всех возможностей электронной библиотеки было использовано довольно много различных JSP страниц, представление всех страниц ниже не представляет смысла. Представим лишь основные страницы разработанного приложения:

1. Начальная приветственная страница, на которую попадает пользователь.

На данную страницу существуют ссылки со всех других страниц



Рис. 10. Начальная JSP-страница

1. Страница регистрации нового пользователя электронной библиотеки

Возможность перехода на страницу регистрации предоставляется только неавторизованным пользователям

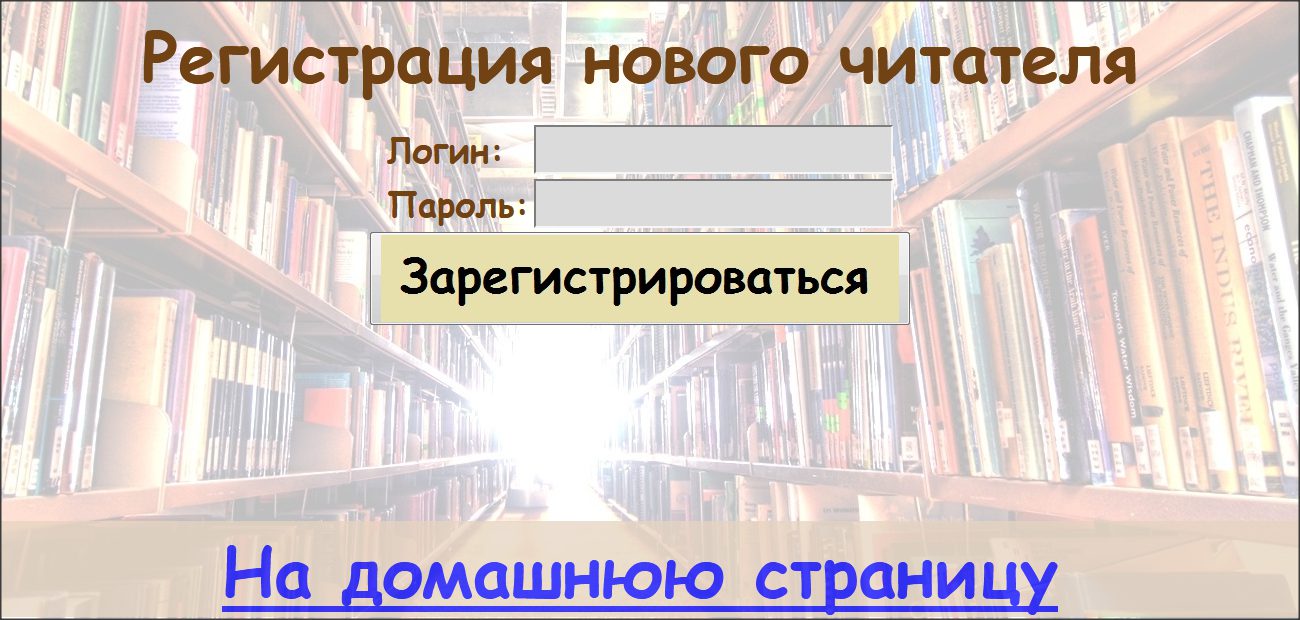


Рис. 11. JSP-страница регистрации нового пользователя

1. Страница авторизации зарегистрированного пользователя

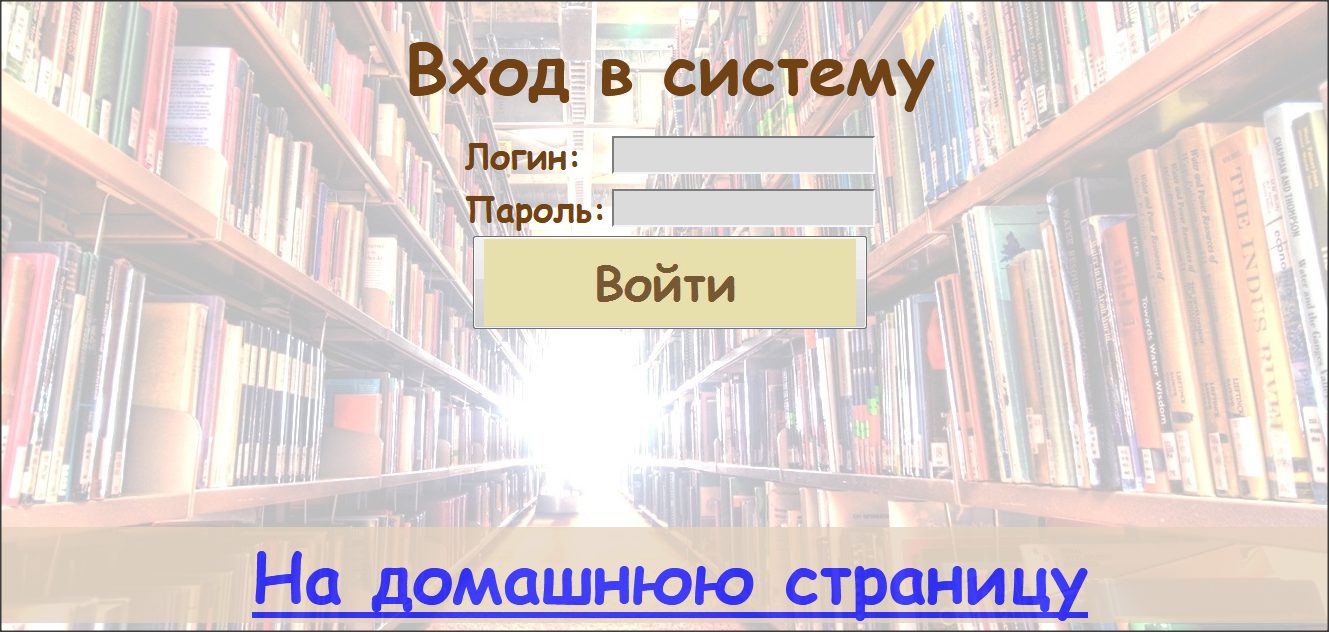


Рис. 12. Страница добавления игрока

1. Начальная страница для пользователя, авторизовавшегося в системе электронной библиотеки.

На данную страницу существуют ссылки со всех других страниц

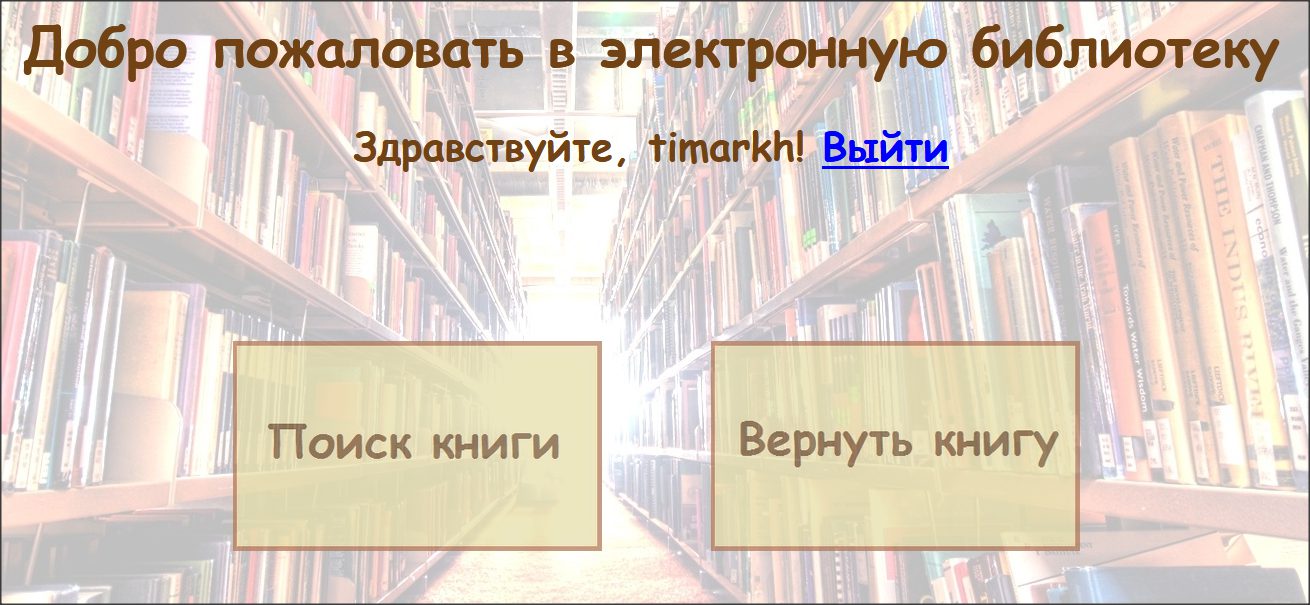


Рис. 13. Начальная страница авторизованного пользователя

1. Страница поиска книги

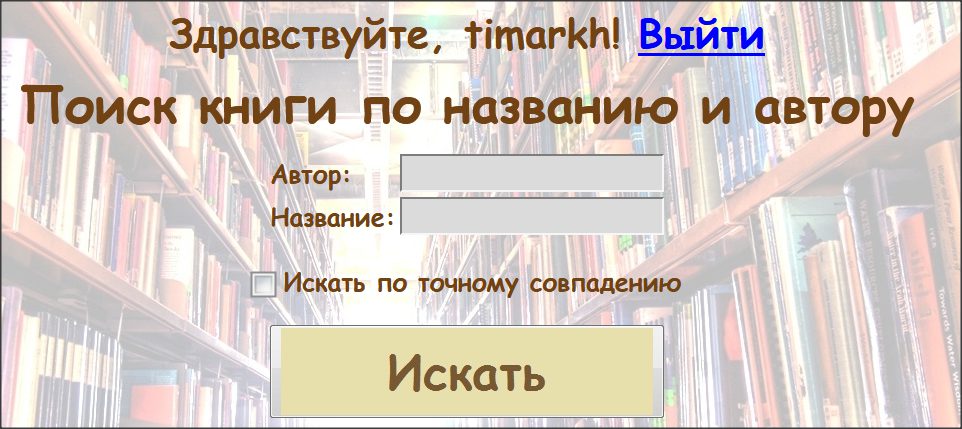


Рис. 14. Страница задания параметров поиска книг

1. Страница с результатом поиска книг по заданным пользователем параметрам



Рис. 15. Страница вывода книг, удовлетворяющих условиям поиска

1. Страница возврата книги авторизованным пользователем



Рис. 16. Страница возврата книги авторизованным пользователем

1. Страница успешного возврата книги в библиотеку



Рис. 17 Подтверждение возврата книги

1. Страница успешного заказа книги – выводится номер заказа

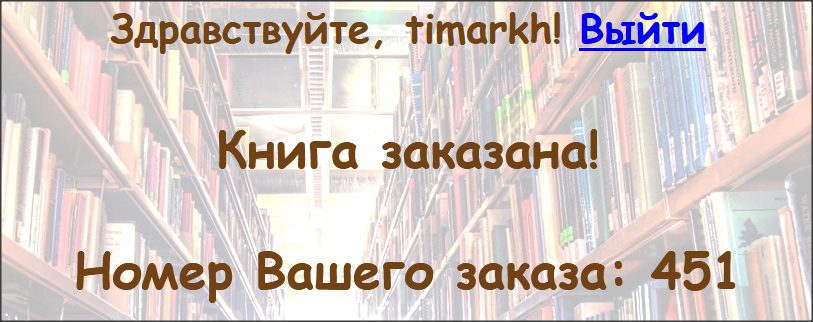


Рис. 18 Подтверждение заказа книги

1. Панель администратора – предлагает простые возможности администрирования системы

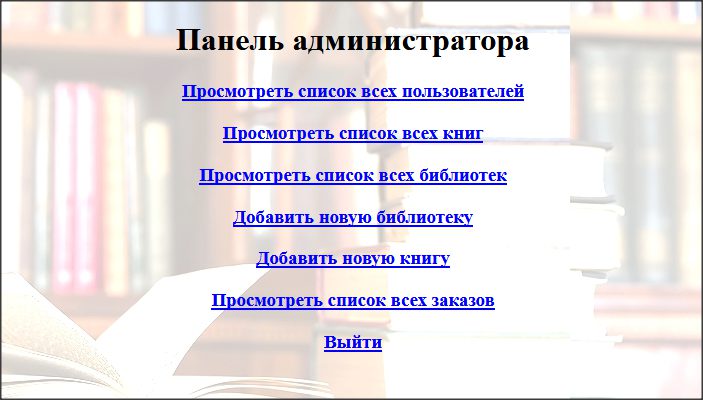


Рис. 19. Панель администратора

## Тестирование

|  |  |  |
| --- | --- | --- |
| **Вариант тестирования** | **Ожидаемый результат** | **Фактический результат** |
| **Пользователь** | | |
| При регистрации пользователь вводит новые логин/пароль | Система должна оповестить пользователя об успешной регистрации и авторизовать пользователя | Система авторизует пользователя, тем самым подтверждая успешную регистрацию |
| При регистрации пользователь вводит уже существующие логин/пароль | Система должна оповестить пользователя о существовании такого пользователя | Выводится сообщение о существовании пользователя с заданным логином |
| При авторизации пользователь вводит корректные логин/пароль | Система должна оповестить пользователя об успешной авторизации и авторизовать пользователя | Система авторизует пользователя, тем самым подтверждая успешную авторизацию |
| При авторизации пользователь вводит некорректные логин/пароль | Система должна оповестить пользователя об ошибках во введенных данных | Система оповещает пользователя о том, где именно (логин или пароль) пользователь допустил ошибку |
| При поиске книг пользователь указывает пустые параметры поиска и ставит флажок «точное совпадение» | Система должна вывести пустую таблицу книг в библиотеке | На экран выводится пустая таблица |
| При поиске книг пользователь указывает пустые параметры поиска и НЕ ставит флажок «точное совпадение» | Система должна вывести таблицу, содержащую все книги из всех библиотек | На экран выводится таблица с книгами |
| При поиске книг пользователь вводит автора и название книги и ставит флажок «точное совпадение» | Система должна вывести только книгу с данным названием (возможно, в разных библиотеках) | Система выводит несколько строк с одной книгой для разных библиотек |
| Авторизованный пользователь нажимает «заказать» напротив одной из книг | Система должна произвести заказ и вывести номер заказа на экран | На экран выводится номер заказа |
| Неавторизованный пользователь нажимает «заказать» напротив одной из книг | Система должна перенаправить пользователя на страницу авторизации/ регистрации, а после успешной авторизации/регистрации произвести заказ книги и вывести номер заказа на экран | Открывается страница авторизации/регистрации, при успехе на экран выводится номер заказа |
| Авторизованный пользователь выбирает возврат книги | Система должна вывести список всех заказанных пользователем книг | На экране выводится таблица, содержащая все заказанные пользователем книги с кнопкой возврата |
| Неавторизованный пользователь выбирает возврат книги | Система должна запросить у пользователя номер заказа, логин и пароль и по ним произвести возврат | После ввода корректных логина/пароля и номера заказа выводится подтверждение возврата |
| При авторизации пользователь вводит **admin** в качестве логина и **adminadmin** в качестве пароля | Система должна авторизовать пользователя как администратора в системе и открыть панель администратора | На экране отображается панель администратора |
| **Администратор** | | |
| Администратор выбирает добавление новой книги в библиотеку | Система должна предупредить администратора о том, что на начальном этапе нет ни одного адреса библиотеки | Система просит сперва добавить хотя бы одну библиотеку |
| Администратор добавляет новый адрес библиотеки | Система вносит новый адрес библиотеки | При открытии списка всех библиотек отображается новый адрес библиотеки |
| Администратор добавляет новую книгу в библиотеку | Система должна добавить новую книгу | При открытии списка всех книг отображается новая книга |
| Администратор добавляет в библиотеку книгу с автором, названием и адресом, уже записанными ранее | Система должна изменить количество доступных для заказа экземпляров книги, описанной администратором | При открытии списка всех книг отображается книга с увеличенным количеством экземпляров |
| Администратор удаляет читателя из системы при просмотре всех пользователей | Система удаляет пользователя и связанные с ним заказы | При просмотре списка пользователей удаленный пользователь не отображается, то же и при просмотре заказов |
| Администратор выходит из системы («разлогинивается») | Система должна открыть начальную страницу с возможностью регистрации, авторизации, поиска и возврата книг | Отображается начальная страница |

## 

## Инструкция системного администратора

Для развертывания приложения необходимы следующие системные компоненты:

* Apache Ant(TM) version 1.9.1
* СУБД JavaDB
* Сервер Glassfish Server 4.0

Настройка СУБД:

* Запустить сервер СУБД через NetBeans, вкладка Services (порт 1527)
* Создать таблицы базы данных с названиями и полями, описанными в разделе «реализация», «слой представления данных»
* Открыть проект через NetBeans
* Нажав на проект правой кнопкой мыши, выбрать Deploy (развернуть), дождаться успешного разворачивания
* Нажав на проект правой кнопкой мыши, выбрать Run (выполнение) для запуска приложения.

После запуска приложения:

* Зарегистрировать в системе пользователя с логином admin и произвольным паролем. При успешной регистрации откроется начальная страница панели администратора.
* Добавить в систему хотя бы один адрес библиотеки
* Добавить в систему все книги, доступные для заказа через систему

## Инструкция пользователя

Пользователю предоставляется интерфейс для взаимодействия с системой.

Изначально пользователь попадает на стартовую страницу, где расположены изображения-ссылки, предоставляющие возможные действия пользователя: регистрация, авторизация, поиск и возврат книги. Если пользователь авторизуется, то ему не предоставляется в дальнейшем возможность регистрации/авторизации.

**Если пользователь авторизуется в системе под именем admin со специфичным для администратора паролем, то происходит перенаправление на панель администратора. В данном случае паролем для администратора был установлен как “adminadmin”.**

## Выводы

В данной курсовой работе получены начальные навыки работы с технологией EJB и сервером приложений Glassfish 4.0, а также разработки WEB приложений при помощи Java Server Pages и взаимодействия с базой данных JavaDB, в среде разработки NetBeans IDE.

Были достигнуты следующие задачи:

* Разработана схема простой базы данных в JavaDB.
* В соответствии с поставленным заданием разработан проект при помощи EJB
* Разработан прототип WEB-приложения на основе jsp-страниц
* Получен опыт взамодействия с сервером приложений Glassfish