EPAM Systems, RD Dep.

Практические задания для тренинга

Task. Web-project

REVISION HISTORY					
Ver.	**Description of Change**	**Author**	**Date**	**Approved**	
				Name	**Effective Date**
<1.0>	Первая версия	Игорь Блинов Ольга Смолякова	<09.04.2014>		

Web-Project

Построить веб-систему, поддерживающую заданную функциональность.

Необходимо написать (построить) веб-систему согласно требованиям, приведенным ниже, и реализовать определенную функциональность.

Общие требования к проекту:

Приложение реализовать применяя технологии Servlet и JSP.

Архитектура приложения должна соответствовать шаблонам Layered architecture и Model-View-Controller.

Информация о предметной области должна хранится в БД:

- данные в базе хранятся на кириллице, рекомендуется применять кодировку utf-8
- технология доступа к БД – JDBC (только JDBC)
- для работы с БД в приложении должен быть реализован пул соединений
- при проектировании БД рекомендуется не использовать более 6-8 таблиц
- доступ к данным в приложении осуществлять с использованием шаблона DAO.

Интерфейс приложения должен быть интернационализирован; выбор языков: английский, русский.

Приложение должно корректно обрабатывать возникающие исключительные ситуации, в том числе вести их журналирование. В качестве логгера использовать Log4j.

Классы и другие сущности приложения должны быть грамотно структурированы по пакетам и иметь отражающую их функциональность название.

При реализации бизнес-логики приложения следует при необходимости использовать шаблоны проектирования (например, шаблоны GoF: Factory Method, Command, Builder, Strategy, State, Observer etc), а также необходимо избегать процедурного стиля программирования.

Для хранения пользовательской информации между запросами использовать сессию.

Для обработки объектов запроса(request) и ответа(response) применить фильтры (например, для установки параметра кодировки запроса/ответа).

При реализации страниц JSP следует использовать теги библиотеки JSTL, использовать скриплеты запрещено. Обязательным требованием является реализация и использование пользовательского тега. Просмотр “длинных списков” желательно организовывать в постраничном режиме.

Документацию к проекту необходимо оформить согласно требованиям javadoc.

Оформление кода должно соответствовать Java Code Convention.

Общие требования к функциональности проекта:

1) Вход(sign in) и выход(sign out) в/из системы.
2) Регистрация.
3) Просмотр информации (например: просмотр всех курсов, имеющихся кредитных карт, счетов и т.д.)
4) Удаление информации (например: отмена заказа, медицинского назначения, отказ от курса обучения и т.д.)
5) Добавление и модификация информации (например: создать и отредактировать курс, создать и отредактировать заказ и т.д.)

Предметные области для разработки проекта.

1. Система **Факультатив**. Существует перечень **Курсов,** за каждым из которых закреплен один **Преподаватель**. **Студент** записывается на один или несколько **Курсов**. По окончании обучения **Преподаватель** выставляет **Студенту** и добавляет отзыв.

2. Система **Платежи**. **Клиент** имеет одну или несколько **Кредитных Карт,** каждая из которых соответствует некоторому **Счету** в системе платежей. **Клиент** может при помощи **Счета** сделать **Платеж**, заблокировать **Счет** и пополнить **Счет**. **Администратор** снимает блокировку.

3. Система **Приемная комиссия**. **Абитуриент** регистрируется на один из **Факультетов** с фиксированным планом набора, вводит баллы по соответствующим **Предметам и аттестату**. Результаты **Администратором** регистрируются в **Ведомости**. Система подсчитывает сумму баллов и определяет **Абитуриентов**, зачисленных в учебное заведение.

4. Система **Библиотека**. **Читатель** имеет возможность осуществлять поиск и заказ **Книг** в **Каталоге**. **Библиотекарь** выдает **Читателю Книгу** на абонемент или в читальный зал. **Книга** может присутствовать в **Библиотеке** в одном или нескольких экземплярах.

5. Система **Больница**. **Врач** определяет диагноз, делает назначение **Пациенту** (процедуры, лекарства, операции). Назначение может выполнит**ь Медсестра (**процедуры, лекарств**а)** или **Врач** (любое назначение). **Пациент** может быть выписан из **Больницы**, при этом фиксируется окончательный диагноз.

6. Система **Турагентство**. **Заказчик** выбирает и оплачивает **Тур (отдых, экскурсия, шоппинг)**. **Турагент** определяет тур как «горящий», размеры скидок постоянным клиентам.

7. Система **Телефонная станция**. **Администратор** осуществляет подключение **Абонентов**. **Абонент** может выбрать одну или несколько из предоставляемых **Услуг**. **Абонент** оплачивает **Счет** за разговоры и **Услуги**. **Администратор** может просмотреть список неоплаченных **Счетов** и заблокировать **Абонента**.

8. Система **Автобаза**. **Диспетчер** распределяет **Заявки** на **Рейсы** между **Водителями,** за каждым из которых закреплен свой **Автомобиль**. На **Рейс** может быть назначен **Автомобиль**, находящийся в исправном состоянии и характеристики которого соответствуют **Заявке. Водитель** делает отметку о выполнении **Рейса** и состоянии **Автомобиля**.

9. Система **Интернет-магазин**. **Администратор** осуществляет ведение каталога **Товаров**. **Клиент** делает и оплачивает **Заказ** на **Товары**. **Администратор** может занести неплательщиков в “черный список”.

10. Система **Авиакомпания**. **Авиакомпания** имеет список рейсов. **Диспетчер** формирует летную **Бригаду** (пилоты, штурман, радист, стюардессы) на **Рейс**. **Администратор** управляет списком рейсов.

11. Система **LowCost-Авиакомпания**. **Клиент** заказывает и оплачивает **Билет** на **Рейс** с учетом наличия\отсутствия багажа и права первоочередной регистрации и посадки (Цена **Билета** может быть ниже стоимости провоза багажа). С приближением даты **Рейса** или наполнением самолета, цена на **Билет** может повышаться.

12. Система **Периодические издания**. **Администратор** осуществляет ведение каталога периодических **Изданий**. **Читатель** может оформить **Подписку**, предварительно выбрав периодические **Издания** из списка. Система подсчитывает сумму для оплаты и регистрирует **Платеж**.

13. Система **Заказ гостиницы**. **Клиент** заполняет **Заявку**, указывая количество мест в номере, класс апартаментов и время пребывания. **Администратор** просматривает поступившую **Заявку**, выделяет наиболее подходящий из доступных **Номеров**, после чего система выставляет **Счет Клиенту**.

14. Система **Жилищно-коммунальные услуги**. **Квартиросъемщик** отправляет **Заявку**, в которой указывает род работ, масштаб, и желаемое время выполнения. **Диспетчер** формирует соответствующую **Бригаду** и регистрирует её в **Плане работ**.

15. Система **Прокат автомобилей**. **Клиент** выбирает **Автомобиль** из списка доступных. Заполняет форму **Заказа**, указывая паспортные данные, срок аренды. **Клиент** оплачивает **Заказ**. **Администратор** регистрирует возврат автомобиля. В случае повреждения **Автомобиля**, **Администратор** вносит информацию и выставляет счет за ремонт. **Администратор** может отклонить **Заявку**, указав причины отказа.

16. Система **Скачки. Клиент** делает **Ставки** разных видов на **Забеги**. **Букмекер** устанавливает уровень выигрыша. **Администратор** фиксирует результаты **Забегов**.

17. Система **Тестирование**. **Тьютор** создает **Тест** из нескольких **Вопросов** закрытого типа (выбор одного или более вариантов из N предложенных) по определенному **Предмету**. **Студент** просматривает список доступных **Тестов,** отвечает на **Вопросы**.

18. Система **Ресторан**. **Клиент** осуществляет заказ из **Меню**. **Администратор** подтверждает **Заказ** и отправляет его на кухню для исполнения. **Администратор** выставляет **Счет**. **Клиент** производит его оплату.

19. Система **Кофе-машина**. **Пользователь** обладает **Счетом**. **Кофе-машина** содержит набор **Напитков**, с заданным числом порций и дополнительных **Ингредиентов**. **Пользователь** может купить один или несколько **Напитков**. **Администратор Кофе-машины** осуществляет ее наполнение.

20. Система **Парк**. **Владелец** парка дает указания **Леснику** о высадке (лечении, художественной обработке, уничтожении) **Растений**. **Лесник** отчитывается о выполнении. **Владелец** просматривает результаты и подтверждает исполнение.

21. Система **Команда разработчиков**. **Заказчик** представляет **Техническое Задание (ТЗ)**, в котором перечислен перечень **Работ** с указанием квалификации и количества требуемых специалистов. **Менеджер** рассматривает **ТЗ** и оформляет **Проект**, назначая на него незанятых **Разработчиков** требуемой квалификации, после чего рассчитывается стоимость **Проекта** и **Заказчику** выставляется **Счет**. **Разработчик** имеет возможность отметить количество часов, затраченных на работу над проектом.