

EXTERNAL JAVA TRAINING. FINAL TASK WEB PROJECT

www.training.by

Legal Notice: This document contains privileged and/or confidential information and may not be disclosed, distributed or reproduced without the prior written permission of EPAM Systems.

November 14, 2020

Final Task. Web Project

- Приложение реализовать применяя технологии Servlet и JSP.
- Архитектура приложения должна соответствовать шаблонам **Layered architecture** и **MVC**. Controller может быть только двух видов: контроллер роли или контроллер приложения.

Требования к БД

- Информация о предметной области должна храниться в БД:
- Если данные в базе хранятся на кириллице, рекомендуется применять кодировку utf-8
- Технология доступа к БД **только** JDBC
- Для работы с БД в приложении должен быть реализован потокобезопасный пул соединений, использовать **synchronized** и **volatile** запрещено.
- При проектировании БД рекомендуется использовать не более 6-8 таблиц
- Работу с данными в приложении осуществлять посредством шаблона DAO.

Базовые требования к приложению

- Интерфейс приложения **должен** быть локализован; выбор из языков: EN|BE|RU etc.
- Приложение **должно** корректно обрабатывать возникающие исключительные ситуации, в том числе вести их логи. В качестве логгера использовать Log4J2/SLF4J
- 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, Singleton, Proxy etc).
- Для хранения пользовательской информации между запросами использовать сессию.
- Для перехвата и корректировки объектов запроса (request) и ответа (response) применить фильтры.
- Разрешается использовать технологии AspectJ и Web-services.
- При реализации страниц JSP следует использовать теги библиотеки JSTL, использовать скриплеты запрещено.
- При реализации пользовательского интерфейса разрешается использовать любые технологии front-end разработки (JavaScript, AJAX).
- Реализовать защиту от повторного выполнения запроса нажатием F5.
- Реализовать собственные теги.
- Просмотр “длинных списков” желательно организовывать в постраничном режиме.
- Валидацию входных данных производить **на клиенте и на сервере**.
- Документацию к проекту необходимо оформить согласно требованиям javadoc.
- Оформление кода **должно** соответствовать Java Code Convention.
- При развертывании приложения разрешается использовать технологию Maven.
- Приложение **должно** содержать тесты TestNG, JUnit, Mockito или EasyMock.
- Приложение **должно** быть размещено на Bitbucket.com

Минимальные требования к функциональности приложения

- Авторизация (sign in) и выход (sign out) в/из системы.
- Регистрация пользователя и/или добавление артефакта предметной области системы.
- Просмотр информации (например: просмотр всех ставок тотализатора, статистики заказов, счетов и т.д.)
- Удаление информации (например: отмена заказа, удаление сущности и т.д.)
- Добавление и модификация информации (например: создать и отредактировать товар, создать и отредактировать заказ и т.д.)
- Во всех проектах где присутствуют финансовые отношения возможно использование оплаты в кредит

Предметные области для разработки проекта

- 1** **Тотализатор.** Клиент делает **Ставки** разных видов на игровой спорт (победа, ничья, поражения, точный результат и пр.) на **Соревнования**. **Букмекер** устанавливает уровень выигрыша. **Администратор** управляет **Пользователями**, создает (изменяет) **Соревнования**, а также фиксирует (random генерация) их результаты.
- 2** **Скачки.** Клиент делает **Ставки** разных видов (победа, первая тройка, аутсайдер, точное место и пр.) на **Скачки**. **Букмекер** устанавливает уровень выигрыша. **Администратор** управляет **Пользователями**, создает (изменяет) **Скачки**, а также фиксирует (random генерация) их результаты.
- 3** **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. **Администратор** управляет Библиотекарями, Читателями и Книгами.
- 4** **Фитнесс-центр.** Клиент осуществляет и оплачивает **Заказ** на цикл тренировок. Тренер определяет нагрузки и режим, делает **Назначение** Клиенту (упражнения, снаряды, питание). **Назначение** может выполнить **Личный Тренер**. Клиент может отказаться и/или заменить часть назначений. По окончании **Заказа** Клиент оставляет отзыв. Постоянным и корпоративным Клиентам назначаются скидки.
- 5** **Интернет-провайдер.** Клиент может просмотреть состояние своего счета и трафика, пополнить счет, изменить личные параметры и тарифный план. **Администратор** регистрирует клиентов, управляет тарифными планами объявляет акции, скидки и т.д. При нарушении правил использования системы может блокировать Клиента. Корректирует тарифные планы и добавляет новые.
- 6** **Подписка.** Администратор управляет Читателями и осуществляет ведение каталога периодических **Изданий** (Газеты, Журналы, Книжные серии). Читатель может оформить **Подписку**, предварительно выбрав периодические **Издания** из списка. Система подсчитывает сумму для оплаты и регистрирует **Платеж**.
- 7** **Заказ АудиоТреков.** Клиент заказывает и оплачивает **АудиоТрек(и)**. Оставляет отзывы об **АудиоТреке**. Администратор добавляет новые **АудиоТреки** и **Альбомы**, формирует **Сборки**, корректирует информацию о существующих, управляет Клиентами, назначая им бонусы, скидки и пр.
- 8** **Пиковая дама (Штосс).** Администратор осуществляет управление Игроками, ведет ленту новостей. Игра происходит как между Игроком и ИИ. Игрок может пополнять свой **Счет**. Игроки могут отправлять сообщения. Победа\поражение изменяют рейтинг Игрока. *Правила: ИИ держит (и метает) банк; он назывался банкометом. Игрок (понтер) делал ставку («куш»). «Понтёры из своих колод выбирают карту, на которую делают*

ставку, и банкومت начинает прометывать свою колоду направо и налево. Если карта понтера легла налево от банкомета, то выиграл понтер, если направо — то банкومت».

- 9 HotelBooking.** Клиент заполняет **Заявку**, указывая количество мест в номере, класс апартаментов и время пребывания. **Администратор** управляет **Клиентами** и **Номерами**, просматривает поступившие **Заявки**, выделяет наиболее подходящий из доступных **Номеров**, выставляет **Счет Клиенту**.
- 10 EPAM-cafe.** Клиент делает **Заказ** на обед (выбирает из меню) и указывает время когда он хотел бы получить заказ. Система показывает цену **Заказа** и предлагает оплатить с клиентского счета или наличными при получении заказа. **Клиенту** за предварительные заказы начисляются баллы лояльности. Если **Клиент** делает заказ, и не забирает его, то баллы лояльности снимаются вплоть до его блокировки. **Клиент** может оценивать каждый **Заказ** и оставлять отзывы. **Администратор** управляет меню, выставляет\убирает баны\бонусы\баллы **Клиентам**.
- 11 Английский Аукцион.** **Администратор** выставляет (снимает, блокирует) на торги лоты. Поддерживается оба вида аукционов: прямой и обратный. **Клиент** может участвовать аукционах, оплачивать покупки, предлагать **Администратору** лоты для продажи. **Администратор** управляет **Клиентами** и **Лотами**.
- 12 HR.** **Соискатель** вакансии регистрируется в системе. **Сотрудник HR** фиксирует результат предварительного собеседования с **Соискателем** и назначает при необходимости техническое собеседование, по результатам которого и принимается решение о трудоустройстве **Соискателя**. **Сотрудник HR** размещает сообщения о вакансиях и управляет ими. **Администратор** управляет **Сотрудниками** и **Соискателями**.
- 13 КиноРейтинг.** **Администратор** создает (управляет) список фильмов, сериалов. **Пользователь** выставляет оценку (один раз) фильму и может оставить отзыв. Его статус автоматически повышается(понижается) если после определенного числа оценок других **Пользователей**, если его оценка близка(далека) от общего рейтинга. **Администратор** управляет пользователями: повышает, понижает статус, ставит баны.
- 14 Такси BUBER.** Клиент делает **Заявку** на вызов такси (координаты вызова указываются автоматически) и указывает место назначения. Система показывает ближайшие свободные машины и предполагаемую цену поездки. Клиент выбирает авто и делает заказ. **Водитель** делает отметку о принятии\выполнении рейса и оплате. **Администратор** регистрирует такси в системе, выдает бонусы постоянным **Клиентам**, выставляет баны **Клиентам** и таксистам. (Координаты **Клиента** и **Такси** генерировать случайным образом)*.

- 15 Прокат велосипедов.** Администратор добавляет велосипеды на разные точки проката. Пользователь берет велосипед в прокат на определенное время. Когда Пользователь возвращает велосипед обратно с баланса списывается стоимость аренды. Администратор управляет велосипедами: создает, удаляет, редактирует; также управляет пользователями: меняет им статус, блокирует.
- 16 Online-Аптека.** Клиент выбирает необходимый препарат из списка доступных. Заполняет форму Заказа, указывая количество и дозировку. Клиент оплачивает Заказ. Фармацевт управляет списком препаратов. Часть препаратов требует электронного рецепта, которые может назначать Клиенту только Врач. Клиент может сделать запрос Врачу на продление рецепта.
- 17 OnlineTraining.** Существует перечень Курсов, за каждым из которых закреплен Преподаватель. Студент записывается на один или несколько Курсов, выполняет задания. Преподаватель выставляет ценки за задания Студенту и добавляет отзыв. Администратор управляет Курсами и Преподавателями.
- 18 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.
- 19 Аукцион Цветов.** Администратор выставляет (снимает, блокирует) на торги лоты. Клиент участвует в аукционах, оплачивает покупки, предлагает Администратору лоты для продажи. Администратор управляет Клиентами и Лотами.
- 20 Конференция.** Администратор создает конференцию, ее секции. Участник регистрируется и подает заявку в одну или более секций. Администратор проверяет заявку на соответствие тематике секции и подтверждает\отклоняет заявку. Участник может снять заявку, задать вопрос Администратору.

REVISION HISTORY					
Ver.	Description of Change	Author	Date	Approved	
				Name	Effective Date
<1.0>	Первая версия	Игорь Блинов Ольга Смолякова	<08-05-2014>		

REVISION HISTORY					
Ver.	Description of Change	Author	Date	Approved	
				Name	Effective Date
<1.1>	Незначительные правки формулировок предложений	Ольга Смолякова	<24-09-2015>		
<1.2>	Изменения общей формулировки задачи. Удаление устаревших вариантов. Добавление новых вариантов	Игорь Блинов	<29-09-2015>		
<1.3>	Добавление нового варианта задания. Уточнение общей формулировки	Игорь Блинов	<06-10-2016>		
<1.4>	Удаление варианта задания. Уточнение общей формулировки и некоторых заданий	Игорь Блинов	<28-11-2016>		
<1.5>	Заменены некоторые задания, откорректированы условия	Игорь Блинов	<10.XI.2016>		
<2.0>	Заменены и откорректированы почти все задания	Игорь Блинов	<07.XII.2016>		
<2.1>	Добавлены два новых задания	Игорь Блинов	<16.II.2017>		
<2.2>	Добавлены 10 задания	Игорь Блинов	<29.VI.2017>		