

Календарь выставок (вар. 7)

Студент:

Алексей Дончак

Тренер:

Любовь Иванова

Описание задачи

Система **календарь выставок**. Существует список *выставочных залов*, в каждом из которых есть перечень *экспозиций*. *Посетитель* покупает *билеты*, оформив *платёж* и выбрав *тему выставки*.

Общие требования (1/4)

Необходимо построить веб-приложение, поддерживающее следующую функциональность:

- На основе сущностей предметной области создать классы их описывающие.
- Классы и методы должны иметь отражающую их функциональность названия и должны быть грамотно структурированы по пакетам.

Общие требования (2/4)

- Информацию о предметной области хранить в БД, для доступа использовать API JDBC с использованием пула соединений, стандартного или разработанного самостоятельно. В качестве СУБД рекомендуется MySQL.
- Приложение должно поддерживать работу с кириллицей (быть многоязычной), в том числе и при хранении информации в БД.
- Код должен быть документирован.

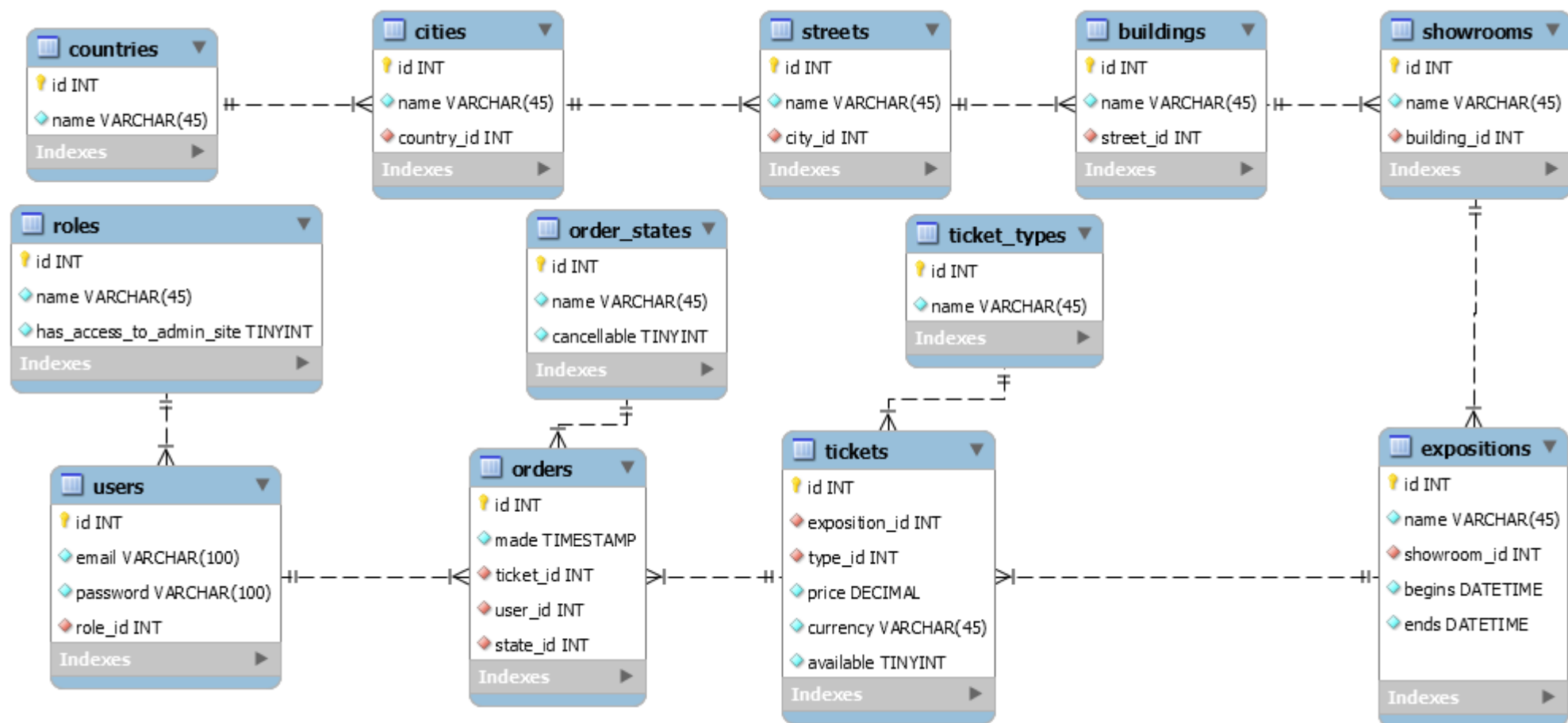
Общие требования (3/4)

- Приложение должно быть покрыто Юнит-тестами.
- При разработке бизнес логики использовать сессии и фильтры, и события в системе обрабатывать с помощью Log4j.
- В приложении необходимо реализовать Pagination, Transaction в зависимости от Вашего проекта.
- Используя сервлеты и JSP, реализовать функциональности, предложенные в постановке конкретной задачи.

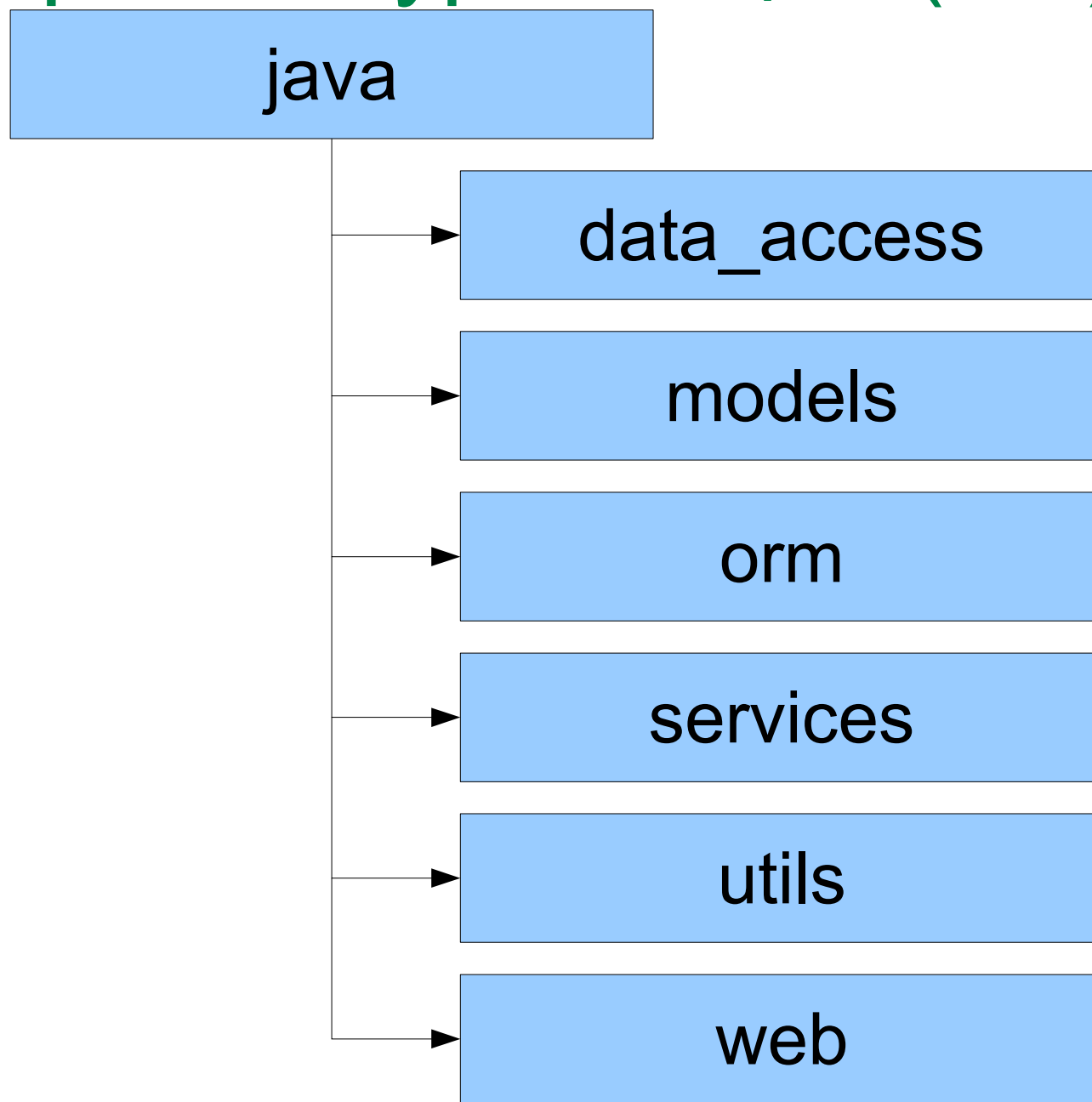
Общие требования (4/4)

- В страницах JSP применять библиотеку JSTL.
- Приложение должно корректно реагировать на ошибки и исключения разного рода (Пользователь никогда не должен видеть stack-trace на стороне front-end).
- В приложении должна быть реализована система Авторизации и Аутентификации.

База данных



Архитектура общая (1/4), пакеты



Архитектура общая (2/4), модели

пакет: models

назначение: хранение информации о сущностях предметной области и связях между ними

логика: минимальна, содержат только данные и могут выполнять их базовую валидацию

ограничения: наследуются от класса Model, для сохранения в БД и интерфейсе администратора должны быть некоторым образом аннотированы

Архитектура общая (3/4), контроллер

пакет: web

назначение: принятие/фильтрация HTTP-запроса и его передача ответственным службам для обработки

логика сервлета: минимальна, сервлет принимает запрос и передаёт его дочерним службам. Вся обработка происходит там.

логика фильтров: установка кодировки UTF-8 и проверка прав доступа к админской части и личному кабинету пользователя

Архитектура общая (4/4), представление

пакет: webapp/jsp

назначение: взаимодействие с
пользователем

логика: отображают данные, проводят
базовую валидацию ввода пользователя

ограничения: не содержат JavaScript.

Pagination тоже выполняет сервер

примечания: повторяющаяся
функциональность реализована через
директиву **include**

Функциональность (1/3), связь с БД

пакет: orm

назначение: отображение полей моделей в таблицы БД и обратно

получение структуры модели: рефлексия

обмен данными с БД: JDBC

ограничения: отображают только заданные типы полей (BooleanField, StringField, ForeignKey<T> и т. д.)

взаимодействие с пользователем: интерфейс IRepository<T> и интерфейсы для запросов

особенность: SqlRepository универсален и не требует переопределения для каждой модели

Функциональность (2/3), особые запросы

пакет: data_access

назначение: взаимодействие с БД за пределами общих CRUD-операций

ограничения: должны быть исполняемы репозиториями

Функциональность (3/3), служебная

пакет: utils

назначение: выполнение разных операций

операции: получение данных из HTTP-запросов и ответов базы данных; обработка метаданных сущностей; выборка локализованных строк; управление транзакциями; установка соединения с БД

Примеры (1/3), сущность «Билет»

```
@Entity(table = "tickets")
@EntityNames(singular = "ticket", plural = "tickets")
public class Ticket extends Model {

    @Column(name = "exposition_id")
    @Relatives(pluralName = "expositions")
    private ForeignKey<Exposition> exposition = new
ForeignKey<>(Exposition.class, false);

    @Column(name = "type_id")
    @Relatives(pluralName = "ticketTypes")
    private ForeignKey<TicketType> type = new ForeignKey<>(TicketType.class,
false);

    private DecimalField price = new DecimalField(false);
    private StringField currency = new StringField(false);

    private BooleanField available = new BooleanField(false);
}
```

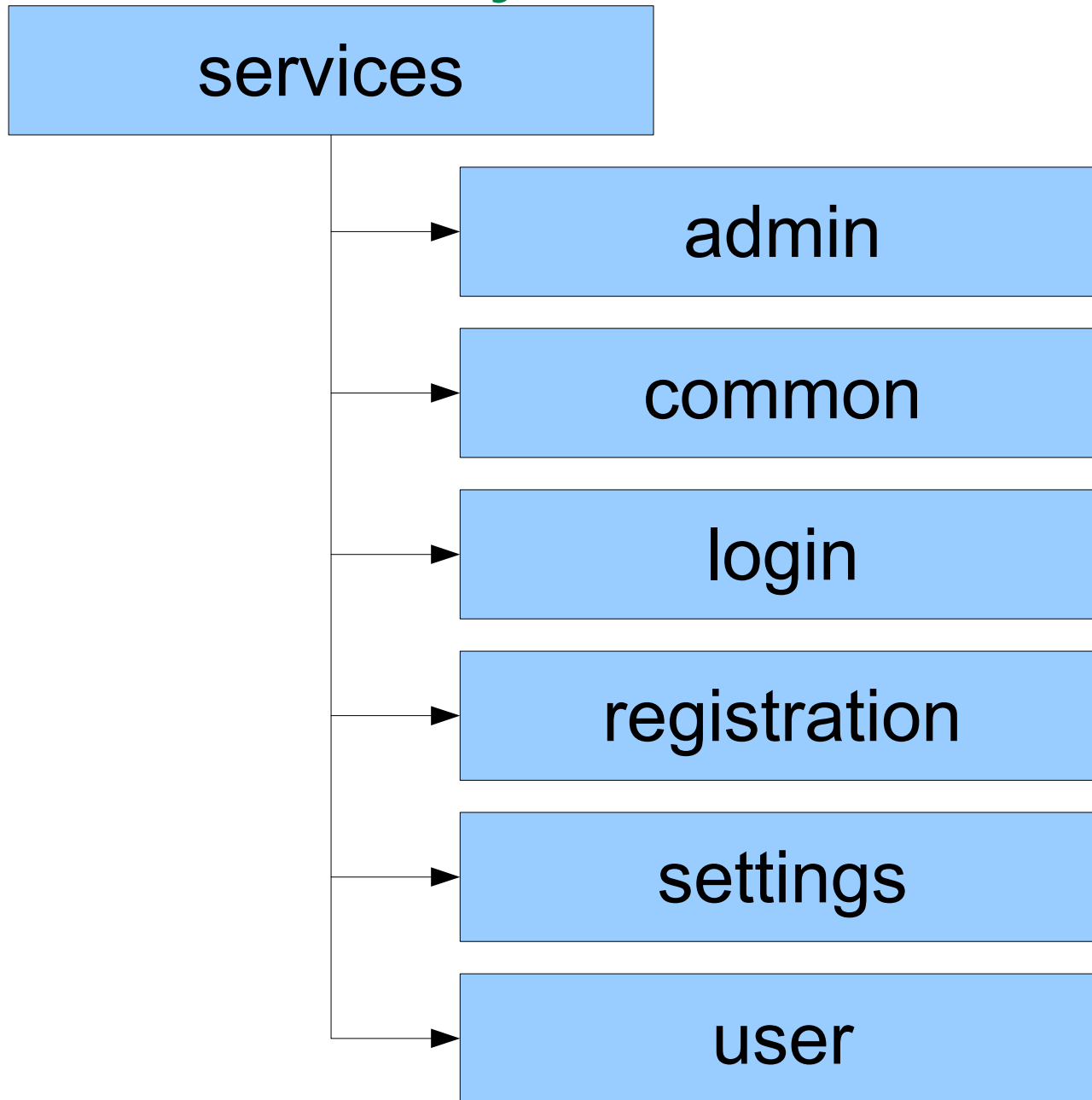
Примеры (2/3), IRepository<T>

```
public interface IRepository<T extends Model> {  
    Optional<T> getById(long id);  
  
    List<T> getAll();  
  
    void delete(T entity);  
  
    List<T> filter(IListEntitiesQuery<T> query);  
  
    Optional<T> get(IGetEntityQuery<T> query);  
  
    void save(T entity);  
  
    Long count();  
  
    Long count(ICountingQuery<T> query);  
}
```


Примеры (3/3), выборка заказов клиента

```
public class FindOrdersByUser extends ListEntitiesQuery<Order> {
    private static final Logger logger =
LogManager.getLogger(FindOrdersByUser.class);
    public FindOrdersByUser withUser(User user) {
        logger.info("started remembering user");
        try {
            statement.setLong(1, user.getId().getValue());
        } catch (SQLException e) {
            logger.error(e);
        }
        logger.info("remembered user");
        return this;
    }
    public FindOrdersByUser(SqlQueryContext<Order> context) {
        super(
            context,
            "SELECT * FROM orders WHERE user_id=?;"
        );
    }
}
```

Службы, пакеты



Администрирование (1/4), описание

пакет: `services.admin`

назначение: CRUD-операции с сущностями предметной области

web-интерфейс: страница администратора со списком классов сущностей; страница с перечислением сущностей класса (pagination присутствует); страница редактирования сущности

примечание: чтобы сущность отображалась корректно, она должна быть зарегистрирована в классе «`utils.managers.MetaInfoManager`» и аннотирована

Администрирование (2/4), список

Календарь выставок

localhost/admin/ticket/show_all











150%

Поиск

Календарь выставок Главная Мои заказы Зона администрации Настройки Выйти

Назад

Таблица

id	price	available	currency	type	exposition	
1	10	1	USD	детский	Выставка цветов	 
2	20	1	USD	взрослый	Выставка цветов	 
4	15	1	USD	детский	Выставка камней	 
5	20	1	USD	взрослый	Выставка камней	 
6	10	1	USD	детский	Выставка картин	 

1 2 Следующая Элементов ▾

Создать

*английские названия полей — плата за универсальность админки

Администрирование (3/4), сущность

The screenshot shows a web browser window with the address bar displaying `localhost/admin/ticket/show_update_form?id=1`. The page has a navigation bar with links: "Календарь выставок", "Главная", "Мои заказы", "Зона администрации", "Настройки", and a "Выйти" button. Below the navigation bar, there are links for "Назад" and "На главную страницу". The main section is titled "Редактирование" and contains a form with the following fields:

- price**: A text input field containing the value "10".
- available**: A dropdown menu with the selected value "Да".
- currency**: A text input field containing the value "USD".
- type**: A dropdown menu with the selected value "детский".
- exposition**: A dropdown menu with the selected value "Выставка цветов".

At the bottom of the form is a blue button labeled "Принять".

*type и exposition — поля типа ForeignKey, подтянулись автоматически

Администрирование (4/4), классы

The screenshot shows a web browser window with the address bar at `localhost/admin`. The page title is "Календарь выставок". The navigation bar includes links: "Календарь выставок", "Главная", "Мои заказы", "Зона администрации", "Настройки", and a "Выйти" button. The main content area is titled "Сущности" (Entities) and contains a list of database entities in a table:

showrooms
ticketTypes
orders
tickets
cities
expositions
userRoles
orderStates
countries
streets
buildings
users

The Windows taskbar at the bottom shows the time as 19:49 on 21.01.2018.

*эта страница тоже генерируется автоматически

Общие службы (1/5), описание

пакет: services.common

назначение: функциональность web-сайта, общая для пользователей, зарегистрированных, незарегистрированных и администраторов

что делают: отображают списки выставочных залов; дают детальное представление по выставкам прошедшим, текущим, запланированным; показывают сведения о билетах

Общие службы (2/5), музеи

Календарь выставок Календарь выставок Календарь выставок +

localhost/common 150% Поиск

Календарь выставок Главная Настройки Регистрация Войти

Выставочные залы

#	Название	Страна	Город	Улица	Номер дома	Выставка		
						Текущие	Старые	Запланированы
1	Первый музей	Украина	Киев	улица Береста	1-а	2	0	2
2	Второй музей	Украина	Киев	улица Егорова	4	0	2	1
3	Третий музей	Украина	Киев	улица Кантарии	8/16	1	1	1



Общие службы (3/5), первый музей

Календарь выставок

Календарь выставок

Календарь выставок

+

localhost/common/search_showroom?id=1150%Поиск

Даз3dfreeThe Neo4j Developer ...LTE - Long Term Evol...LTE Architecture and i...C'est la vie, Lily - Joe D...Getting Started - Acces...Getting Started - Acces...Joseki at Sensei's LibraryEt l'amour s'en va - Jo...Le château de sable - J...Que sont devenues m...L'albatros - Joe Dassin ...

Календарь выставокГлавнаяНастройки

РегистрацияВойти

Первый музей

EN

20:1021.01.2018

Общие службы (4/5), выставка цветов

Календарь выставок Главная Настройки Регистрация Войти

Выставка цветов

Название:	Выставка цветов
Выставочный зал:	Первый музей
Начинается:	2018-01-02
Заканчивается:	2020-01-03
Билеты:	<div>детский, USD, 10</div> <div>взрослый, USD, 20</div>



Общие службы (5/5), детский билет

Календарь выставок Главная Настройки Регистрация Войти

Информация о билете

Выставка:	Выставка цветов
Адрес:	Украина, Киев, улица Береста, 1-а
Начинается:	2018-01-02
Заканчивается:	2020-01-03
Тип:	детский
Цена:	USD, 10

Войдите в учётную запись чтобы приобрести билет.

*пользователь не может без авторизации приобрести билет

Авторизация (1/4)

пакеты: services.login, services.registration

назначение: контроль доступа к разным частям приложения

примечание: администратор тоже является пользователем и может приобретать билеты

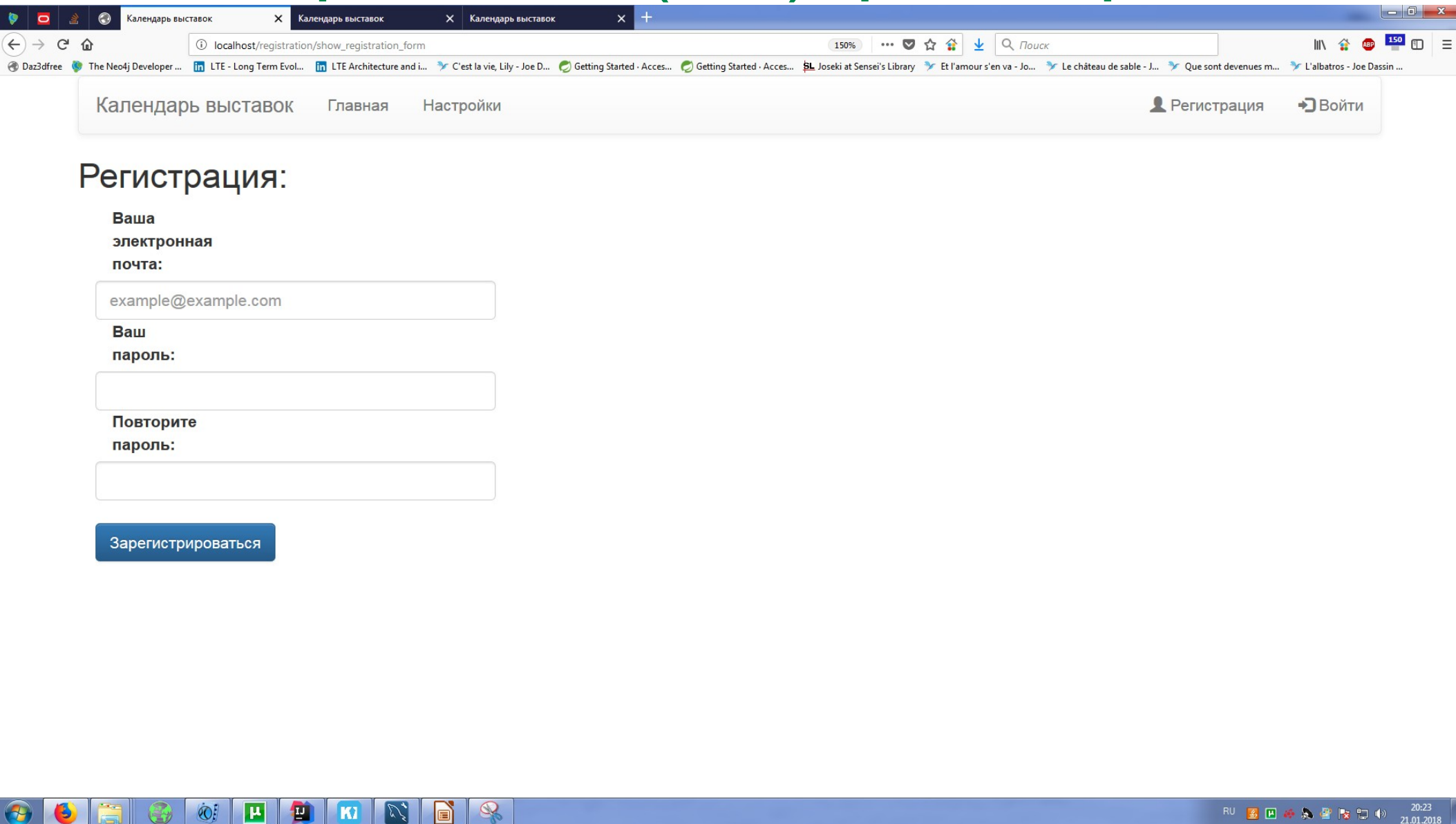
защита: пароли шифруются алгоритмом AES

ограничения: могут быть сколько угодно пользователей с одинаковой почтой, но разными паролями

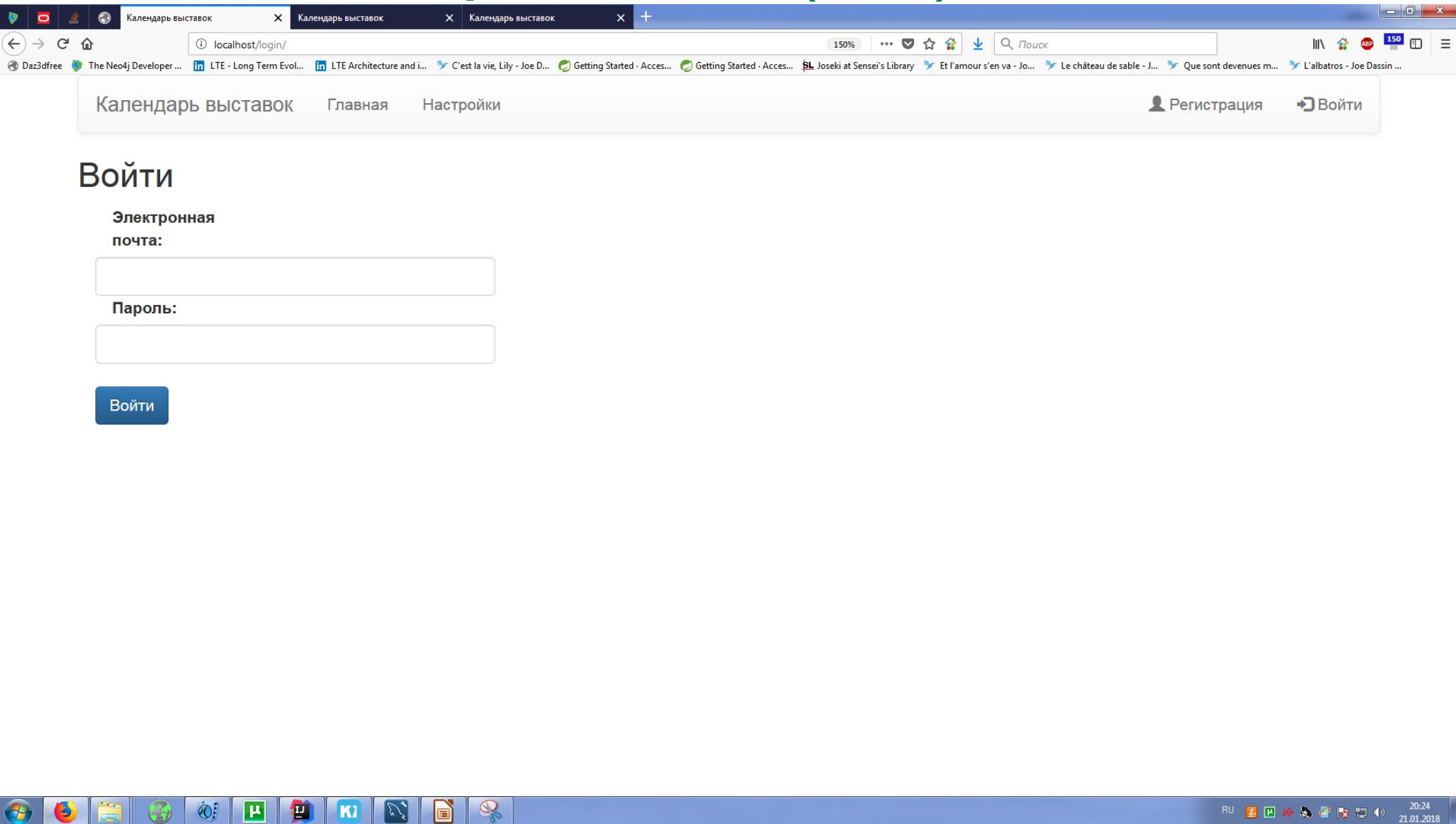
работа: в объекте сессии хранится экземпляр типа User

ВОЗМОЖНОСТИ: после авторизации пользователь может приобретать билеты, просматривать заказы и отменять их

Авторизация (2/4), регистрация



Авторизация (3/4), вход



Авторизация (4/4), заказы

The screenshot shows a web browser window with the URL `localhost/user/orders`. The page has a navigation bar with links: "Календарь выставок", "Главная", "Мои заказы", "Зона администрации", and "Настройки". A "Выйти" button is in the top right. The main section is titled "Мои заказы" and contains a table of orders.

#	Выставка	Билет	Цена	Приобретён	Статус	
1	Выставка цветов	взрослый	USD, 20	2018-01-03	новый	
2	Выставка цветов	детский	USD, 10	2018-01-20	отменён	
3	Выставка цветов	детский	USD, 10	2018-01-20	отменён	
4	Выставка цветов	детский	USD, 10	2018-01-21	отменён	
5	Выставка цветов	взрослый	USD, 20	2018-01-21	отменён	

*неадминистратор не увидит кнопку «Зона администрации»

Настройки (1/2), описание

пакет: services.settings

назначение: пользовательские настройки

ограничения: доступна только смена языка;
динамическая часть админки не локализована;
также не затрагиваются строки, хранящиеся в
БД; информация не выходит за рамки сессии

локализация: все строки, видимые
пользователями, хранятся в **interface.properties**

Настройки (2/2), страница

