Тестовое задание

Цель:

Мы хотим понять на сколько хорошо вы можете разобраться с новыми технологиями в короткий срок, и в целом понимание отдельных архитектурных и технических подходов в части серверной разработки.

Описание:

Наша предметная область - это оказание государственных услуг в электронном виде гражданам страны, большая часть услуг могут быть получены в любой момент и не имеют ограничений по количеству человек, которые могут ее получить, но есть категория «сезонных» услуг, выделяющихся на фоне остальных ограниченным количеством. Например, сезонная выдача охотбилетов, запись ребенка в первый класс и т.д. Момент когда «сезонные» услуги становятся доступны для оказания всегда характеризуется высокой нагрузкой на систему и большим количеством одновременных пользователей пытающихся получить услугу. Количество желающих всегда больше чем доступно к оказанию.

Нужно написать веб приложение упрощенного оказания «сезонной» услуги.

Со стороны пользователя доступны следующие действия:

1. Форма оказания услуги
2. Список всех оказанных услуг
3. Карточка с информацией об оказанной услуге

Начальная страница «Список всех оказанных услуг»

Форма оказания услуги:

1. Страница с формой для заполнения (Фамилия, Имя, Отчество, e-mail при желании можно расширить этот список) и кнопками «отправить» и «отмена»
2. После того как пользователь заполняет форму и нажимает отправить, получает сообщение об успешном заполнении и возможность перейти в карточку с подробной информацией или посмотреть весь список оказанных услуг
3. В случае «Отмена» возвращается к списку всех оказанных услуг

Список всех оказанных услуг:

1. Содержит таблицу всех оказанных услуг (список колонок: дата создания, номер, наименование услуги) список отсортирован по дате создания (вверху списка отображаются новые услуги)
2. Из каждой строки можно перейти в подробную карточку оказанной услуги
3. На странице находится кнопка «Получить услугу» - ведущая на «Форма оказания услуги»

Карточка с информацией об оказанной услуге:

1. Содержит информацию об оказанной услуге (ФИО, дату оказания, номер, название услуги)
2. Кнопку вернутся к «Список всех оказанных услуг»

Процесс оказания «сезонной» услуги

Каждая услуга должна содержать информацию:

1. Наименование
2. Количество сколько раз ее можно оказать (ориентировочный объем данных от 5000 до 10000 раз)

Процесс оказания услуги

1. В момент сохранения формы, указать дату создания, присвоить порядковый номер заявления и вернуть его пользователю
2. Определить успел ли пользователь подать заявление и попал в отведенный лимит по услуге
3. Если пользователь попал в лимит отправить уведомление на e-mail об оказании услуги, если не попал отправить письмо об отказе
4. Процесс уведомление по e-mail, должен обеспечивать гарантированную доставку письма от приложения до сервера SMTP, в случае если сервер SMTP не доступен повторить попытку позже.
5. Будем считать в базовом варианте задания доступна для оказания только одна услуга.

Дополнительным плюсом будет реализация следующих функций

1. **Реализовать клиентскую часть –** подробнее в блоке «Возможные допущения»
2. **Авторизация по логину и паролю и система ролей –** дать возможность заходить и заполнять форму только авторизованным

пользователям, дополнительно в общем списке услуг добавить фильтрацию общего списка по авторизованному пользователю + форму логина (по желанию данную тему можно развить в части форма регистрации, сброса пароля с уведомлением на e-mail и т.д.)

3. **Шаблоны писем –** расширить механизм уведомления через e-mail в виде html писем
4. **Оказание нескольких «сезонных» услуг одновременно –** расширить форму получения услуги выпадающим списком с возможностью выбрать услугу.

Важно!

Общие пожелания

1. Добавьте к решению инструкции по запуску проекта и краткое описание решения
2. Опубликовать исходный код, например на GitHub

Приложение должно быть построено на следующем стеке технологий

1. **REST API** - взаимодействие клиент – сервер
2. **PostgresSQL** - реляционная база данных
3. **Spring/JavaEE –** основа веб приложения
4. **Hibernate –** реализация JPA
5. **Maven –** сборщик проекта
6. **Tomcat/Wildfly –** сервер приложения
7. **Java 8**

Возможные допущения:

1. В случае если базовых знаний клиентской разработки нет или в целом это не интересно, допускается замена клиентской части документацией по контракту между клиентом и сервером (REST API с примерами и последовательность вызовов для каждой страницы и их сценариев).
2. В качестве интеграции с SMTP сервером можно выбрать любой (gmail, yandex и т.д.), в том числе fakesmtp

Критерии оценки:

1. Архитектурные решения в условиях большой интенсивности и объемов данных
    a. для подачи заявлений и принятия решений о попадании в лимит
    b. отображение списка оказанных услуг
    c. организация отправки уведомлений на email

2. Качество кода
3. Проектирование реляционных баз данных
4. Организация процесса логирования в приложении и места использования
5. Организация работы с настройками и перечень того что вынесли (настройки базы, e-mail)
6. Валидация входных данных и граничных условий.
7. Структура проекта
8. Работа с Git чтобы понять какими этапами шла разработка (коммиты, сообщения к ним) (плохой пример один initial commit)
9. Написание Unit тестов + теория тестирования (выделение тест кейсов, классов эквивалентности, AAA pattern, feedback от части asserts). **Данный пункт является огромным плюсом.**