Итоговое задание

В задании требуется написать приложение, моделирующее работу информационной системы автоматизации документооборота медицинского реабилитационного учреждения. Ниже более подробно описаны предметная область и технические требования.

# Предметная область

Существуют следующие виды сущностей:

* Пациент
  + Имя
  + Диагноз
  + Номер страховки
  + Лечащий врач
  + Статус (лечится/выписан)
* Назначение
  + Пациент
  + Тип (процедура/лекарство)
  + Временной паттерн (дважды в день – утром и вечером, например)
  + Период
  + Доза (если лекарство)

Пример назначения процедуры: физиотерапия 2 раза в неделю по вторникам и четвергам.

Пример назначения лекарства: давать аспирин 2 раза в день утром и вечером.

* Событие
  + Пациент
  + Дата и время
  + Статус (запланировано/выполнено/отменено)
  + Процедура/лекарство

Пример: запланировано дать Иванову 1 таблетку аспирина 10 марта 2019 в 9:00.

* Процедуры и лекарства
  + Название
  + Тип

Приложение должно обеспечивать следующую функциональность:

* Для врачей клиники
  + Добавление пациента
  + Выписка пациента (подразумевает завершение всех назначений с момента выписки)
  + Назначение процедур и лекарств
  + Редактирование назначений (изменение периода, дозы, паттерна)
  + Отмена назначений

При создании назначения необходимо сгенерировать все предусмотренные им события. Например, в случае с назначением делать физиотерапию 2 раза в неделю по вторникам и четвергам с периодом на 2 недели должно сгенерироваться 4 события.

При отмене назначения, все ещё не выполненные предусмотренные им события должны быть отменены.

* Для медсестёр клиники
  + Просмотр всех событий.
  + Фильтрация событий по дате (на сегодня, на ближайший час) и по пациенту.
  + Изменение статуса событий из «запланировано» в «выполнено» и из «запланировано» в «отменено».

При переводе события в статус «отменено» медсестре необходимо указать причину отмены: пациент плохо себя чувствовал, не было технической возможности выполнить назначение и т.д.

# Технические требования

В итоге требуется получить многопользовательское приложение типа клиент-сервер с соединением по сети.

Все данные хранятся на стороне сервера. Каждый клиент может загружать некоторые данные, после каждой операции изменения данные должны быть синхронизованы с сервером.

Клиент должен иметь графический интерфейс.

Приложение должно обрабатывать аппаратные и программные ошибки.

# II часть

Реализовать отдельное клиент-приложение для электронного табло. Приложение должно отображать список всех событий, назначенных на текущий день. Данные должны подгружаться при старте и храниться на стороне клиента. Перезагрузка данных осуществляется в случае получения уведомления от сервера об изменениях в списке событий (добавлены новые или отменены старые).

# Критерии успешного выполнения

1. Функциональность работает (обязательно наличие UI)

2. Maven-based проект, разбитый на модули (билд одной командой, деплой одной командой)

3. Описаны интерфейсы предметной области

4. Подключена БД MySQL

5. Созданы сущности предметной области; маппинг на таблицы в БД

6. Работа с сущностями через DAO

7. Приложение развернуто на AS

8. Реализована обработка исключений

9. Подключено логгирование

10. Наличие technical solution description

11. Наличие unit-тестов на бизнес логику

# Используемые технологии

В рамках школы студенты не ограничены в выборе технологий для реализации проекта в рамках стека технологий, используемых в компании в различных проектах (лекции покрывают небольшую часть из них). Список технологий для выбора приведен ниже. Однако проект должен быть написан преимущественно на java (допускается написание незначительных частей, отдельных модулей или дополнительной функциональности на других языках, например, Kotlin или Python). После обсуждения с куратором и по согласованию с ним студент может выбрать любые комбинации технологий и фреймворков, но стоит учитывать, что куратор может быть незнаком с какими-то из них.

Список технологий, используемых в некоторых проектах компании:

Spring Core,Spring Data,AOP,Spring Boot,Spring Security,MVC,Spring Cloud, Microservices, Data processing(Spark, Apache Flume), CDI/EJB, Apache CXF, RedHat Fuse, Oracle eCommerce (ATG, Endeca...), JBoss, Apache Tomcat, Embedded Tomcat H2 (in-memory), Websphere Application Server 9.0, React, Angular/NPM/Webpack, typescript, JSF/ExtJs, JSP/Servlets, JavaScript / TypeScript / HTML / Sass / CSS/LESS/jQuery/GraphQL, SOAP/REST, JDBC, XML/XSLT/XSD,Bash scripting, Oracle Database, Gradle, Git, Maven, Subversion, Jenkins, Grafana, GitlabCI, Sonar, UNIX shell, Bootstrap, Formbased authorisation/JWT, Drools, ARS, DOM, PL SQL, Oracle AQ, IBM MQ, JMS, OracleDB ,Apache Camel, Docker, Microsoft T-Sql, WebDriver (Java), Oracle ATG, python, mapstruct, Lombok, Groovy, openshift, EC2, S3, Mongo, Casandra, hazelcast ,Junit, testing, Selenium, Mockito, Selenid, Prometheus, Kubernetes , Helm, Kibana, AWS, Apache Camel, Go, K8s, ETCD, PostgreSql, Wicket, blockchain, Web Flux, JPA/Hibernate

Задание состоит из 2 частей, каждая часть должна быть реализована с использованием разных неповторяющихся технологий.

Предлагаемый вариант стека технологий для 1 реализации части на основании лекций:

* IDE - Any (Eclipse, IDEA)
* Tomcat
* DB – MySQL
* Maven
* JPA
* Spring Framework (кроме Boot, Data)
* JSP

Для 2 части:

* Maven
* AS – WildFly
* EJB
* JSF
* MQ (для уведомлений от сервера)
* WebServices (для обмена данных между клиентом и сервером)

Подключение любого фреймворка/библиотеки принимаются в зачет только при условии выполнения пунктов, описанных выше.

Рекомендуем использовать последнюю стабильную версию технологий, обязательных к использованию.

Плюсом будет использование следующих технологий: Selenium, Sonar, Angular/React (например, админка для табло), Docker, Microservices, использования доступного cloud для развертывания приложения (например, <https://cloud.yandex.ru/> - 1 месяц бесплатного использования) и/или наличие «killer features».