# ВВЕДЕНИЕ

Проблема образования в нашем мире является одной из самой обсуждаемых. Многие родители хотят, чтобы их дети получили релевантное образование, но не знают, каким образом можно повлиять на ребенка и стимулировать его к обучению.

Последнее и было главной задумкой при создании дипломного проекта. Стимулирование через интересные задания, красивое оформление и оплату мобильного телефона должны привлечь детей к обучению, а автоматизация проверки и ведение статистики позволят родителям наблюдать за успехами детей не тратя время на проверку и составление тестов.

Конечно, для удобства задания распределены по темам, а темы по предметам. В свою очередь у каждого предмета имеется атрибут «Класс» («schoolclass»). Этот атрибут был введен для регулирования сложности заданий и тем относительно друг друга. На данный момент программа содержит предметы для 11 различных классов, что соответствует российской учебной программе вплоть до полного среднего образования. Несмотря на всё это диапазон классов не является статичным и может быть расширен в случае, например, введения программы в пользование в стране с иной системой образова

# ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

## Организация ввода-вывода и программного интерфейса

### Входные и выходные данные

Входными данными приложения являются HTTP-запросы, а выходные, соответственно, HTTP-ответы в соответствии со спецификацией RFC 2616 HTTP/1.1.

Внимание! Для использования кросс-доменных запросов убедитесь в том, что приложение настроено для их принятия и дает ответ с соответствующими заголовками ответа (response headers) на предварительный запрос методом OPTIONS.

Приложение может использовать в качестве данных для обработки такие части запроса, как переменные путей, параметры запроса (path variables и request parameters, представляющие из себя query string) и тело запроса (request body).

Сервер приложений принимает строку, определяет контекст к которому сделан запрос, соотносит это с приложением, которое находится в этом контексте и направляет его соответствующему сервлету (сервлеты объявлены в web.xml). Данное приложение построено на основе Spring MVC Framework, так что в данном случае управление по выбору обработчика обычно предоставляют сервлету dispathcerServlet (класс org.springframework.web.servlet.DispatcherServlet) см. Рис 1.

DispatcherServlet

Request

Response

Handler Mapping

RestController

Рис 1. Схема обработки запроса внутри приложения Spring MVC

Как можно увидеть из вышеприведенной схемы, DispatcherServlet является посредником между приложением основанном на технологии Spring MVC и сервером и занимается решением задач связанных с обработкой запроса. Так, например, в случае нашего приложения, где каждая функция обработки помечена аннотацией RestController, автоматически результат функции будет помечен как ResponseBody, что является поводом для преобразования возвращаемого объекта в JSON библиотекой jackson (или в любой другой медиа-тип, обозначенный в параметре produces аннотации RequestMapping). Если же не будет найдено аннотации RestController у класса, или ResponseBody у результата функции, то DispatcherServlet, по умолчанию действующий в соответствии с шаблоном MVC постарается обратиться к стандартной реализации ViewResolver и найти представление, сооответствующее возвращаемому String, Model, или ModelAndView значению. Как можно заметить из описания, сервлет-диспетчер действительно говорящее название для данного класса.

### Взаимодействие с базой данных

Приложение по умолчанию пытается найти базу данных в контексте ресурсов сервера по имени «java:/PostgresDS» (технология JNDI). Для того чтобы изменить способ подключения необходимо в конфигурации