Московский государственный технический университет

имени Н.Э. Баумана

Факультет «Информатика и системы управления»

Кафедра «Программное обеспечение ЭВМ и

информационные технологии»

**Техническое задание по проекту**

**«Система проверки и выдачи заданий по иностранному языку»**

Пахомов А.А., студент ИУ7-27

Москва, 2017

## Глоссарий

|  |  |
| --- | --- |
| **Термин** | **Определение** |
| Валидация  данных | Проверка на корректность, полноту и непротиворечивость входных, выходных и обрабатываемых данных |
| WEB-интерфейс | Интерфейс пользователя, предоставляемой системой через Web-браузер. В разрабатываемой системе только один веб-интерфейс. |
| СОА (SOA) | Сервис-ориентированная архитектура (Service Oriented Architecture), [модульный](https://ru.wikipedia.org/wiki/Модульность_(программирование)) подход к разработке [программного обеспечения](https://ru.wikipedia.org/wiki/Программное_обеспечение), основанный на использовании [распределённых](https://ru.wikipedia.org/wiki/Распределённые_вычисления), [слабо связанных](https://ru.wikipedia.org/w/index.php?title=Слабое_связывание&action=edit&redlink=1) заменяемых компонентов, оснащённых стандартизированными [интерфейсами](https://ru.wikipedia.org/wiki/Интерфейс_программирования_приложений) для взаимодействия по стандартизированным [протоколам](https://ru.wikipedia.org/wiki/Сетевой_протокол). |
| Профиль пользователя | Информация о пользователе портала, хранящаяся в портале, в частности, имя, фамилия, фотография и др. |
| Сессия | Сессия на сайте - серия запросов к порталу, сделанных одним пользователем в заданный промежуток времени, в данном документе - в течение 30 минут. |
| Фронтенд | Серверное приложение, принимающее запросы от пользователя портала. На каждый из типов запросов от пользователя (показать новости, показать список пациентов и др.) фронтенд определяет, как организовать выполнение запроса. Фронтенд принимает запросы от пользователя, анализирует их и в соответствии с заложенным алгоритмом выполняет запросы к бекендам. |
| Бэкенд | Серверное приложение, выполняющее определенную задачу, например, взаимодействие с СУБД. Бекенды принимают запросы от фронтенда. |
| Сервер | Компьютер, выполняющий функции обслуживания пользователей при доступе к информационным ресурсам в вычислительных системах. |
| Шардинг | Разделение данных по диапазонам первичных ключей между несколькими базами данных. |
| Проект, портал, система | В данной работе термины «проект», «портал» и «система» взаимозаменяемы. |

## 

## Раздел 1. Общие сведения

Данное техническое задание составлено для разработки проекта «Система проверки и выдачи заданий по иностранному языку». Техническое задание выполнено на основе ГОСТ 19.201—78 «ЕСПД. Техническое задание. Требования к содержанию и оформлению».

Разработка ведется в рамках выполнения лабораторных работ по курсу Методология программной инженерии на кафедре «Программное обеспечение ЭВМ и информационные технологии» факультета «Информатика и системы управления» МГТУ им. Н. Э. Баумана.

## Раздел 2. Назначение и цели создания системы

!!!уточнить и конкретизировать!!!

словари, подсказки и тп.. аудио, перевод, звук или текст и т.п…

Различные студенты мыслят по-разному и делают различные ошибки при выполнении домашних заданий. В частности, если рассмотреть задания по иностранному языку, можно отследить ряд закономерностей. Например, какие ошибки чаще делают испанцы при обучении русскому языку, или, скажем, какие ошибки делают русские при изучении английского и чем эти ошибки отличаются от тех, которые делают глухонемые студенты. Разрабатываемая система призвана помочь преподавателю собрать и визуализировать эти ошибки и классифицировать закономерности.

Данное техническое задание определяет требования к разработке информационной системы наглядного представления о выполнении домашних заданий по иностранному языку, анализе их ошибок и общих тенденций в зависимости от категории студента.

## Раздел 3. Требования к системе

!!!вынести в глоссарий!!!

сначала основные требования! географическое распределение,

кол-во серверов, какие под бэкап, объём информации и т.п.

1. Все взаимодействие выполняется по rest, желательно использовать нотацию RESTful.
2. Связь пользователей между системами выполняется с помощью UUID.
3. Все системы должны собираться и деплоиться через CI (на локальной машине в случае развертывания CI на локальной машине, либо на какой-либо IaaS типа Digtal Ocean или Heroku в случае использования Travis CI). Так же желательно использовать контейнеризацию Docker или Vagrant (Docker предпочтительнее).
4. Можно использовать SQL или NoSQL базы. Каждая система имеет свою собственную базу, прямой запрос к сторонней базе запрещен.
5. Предусмотреть возможность масштабирования сервисов.
6. Использовать Git и GitHub.

**Требования к пользовательскому интерфейсу**

1. Получение задания студентом
2. Решение домашнего задания студентом
3. Отправка выполненного задания
4. Интерфейс для преподавателя для просмотра и анализа выполненных заданий

**Требования к надежности**

Необходимо предусмотреть ситуацию недоступности систем, обработку таймаутов и ошибок сервисов. В случае ошибки/недоступности некритичного функционала выполнять деградацию функциональности.

## Раздел 5. Порядок контроля и приёмки системы

1. Показать процесс сборки проекта из master’а и деплоя.
2. Продемонстрировать работоспособность в случае доступности всех сервисов (happy way) и поведение системы в случае недоступности одного или нескольких сервисов.
3. Отдельно продемонстрировать обработку медленного ответа сервиса.
4. Продемонстрировать возможность масштабирования сервисов (устное объяснение, без самой демонстрации).
5. Расчетно-пояснительная записка с диаграммами, описывающими функционирование системы.