1. **ВВЕДЕНИЕ**

Веб-приложение «The System» является спец приложением для контроля выполненных задач, что делает приложение актуальным на сегодняшний день. Поскольку большинство компаний, учебных заведений уходят на дистанционную работу отделы информатизации, не способен обрабатывать удаленные запросы.

Разработанное приложение повысит эффективность работы отдела информатизации, а также позволит создавать запросы из любых мест при наличии доступа к интернету.

«The System» предлагает создание задачи для отдельного пользователя или для всех пользователей. Также позволяет хранить изображение и чат к каждой отдельной задачи.

1. **ОСНОВНАЯ ЧАСТЬ**
   1. **Описание программы**

Разработка веб-приложения предназначена для построения задач отделам информатизации или в другой сфере, где требуется постановление задач между несколькими пользователями.

* 1. **Функциональные возможности**

The System предоставляет:

* + Создание пользователей
  + Создание задач
  + В каждой задаче имеется чат
  + Загрузка изображений в чате и в задаче
  1. **Требование к программе**

The System состоит из двух частей клиентская и серверная. На клиентской выводится графический интерфейс, а на серверной происходит обработка.

Клиентская часть представляет из себя набат веб страниц, с различными типами данных.

Серверная часть представляет из себя сервер, связанный с БД и файловой системой, на которой хранятся файлы.

* 1. **Структура приложения**

Приложение имеет следующий вид в папке src находится две папки main и test, в main хранится java и resources. В java находится код программы, в resources файлы с настройкой проекта и веб страницы.

* 1. **Классы и компоненты**

Классы и компоненты представлены в виде UML.

Во вторичную часть входят папки domain, controller, repo, services, config, model:

* Domain хранит описание БД
* Controller отвечает за прием данных от клиента по URL запросам
* Repo предоставляет запросы к БД
* Services отвечает за основную обработку логической части приложения
* Config настройка основных действий приложения
* Model содержит DTO и базовую проверку правильности входных данных

На диаграмме классов Domain на рисунок 1 представлена реализация БД через hibernate. Класс User отвечает за хранение данных пользователей и имеет доступ классу Roles через hibernate в котором хранятся роли пользователей. Класс   
 Massages хранит сообщения пользователей, а класс Img хранит название изображенией и их место на диске. А Enum хранят базовые типы элементов.

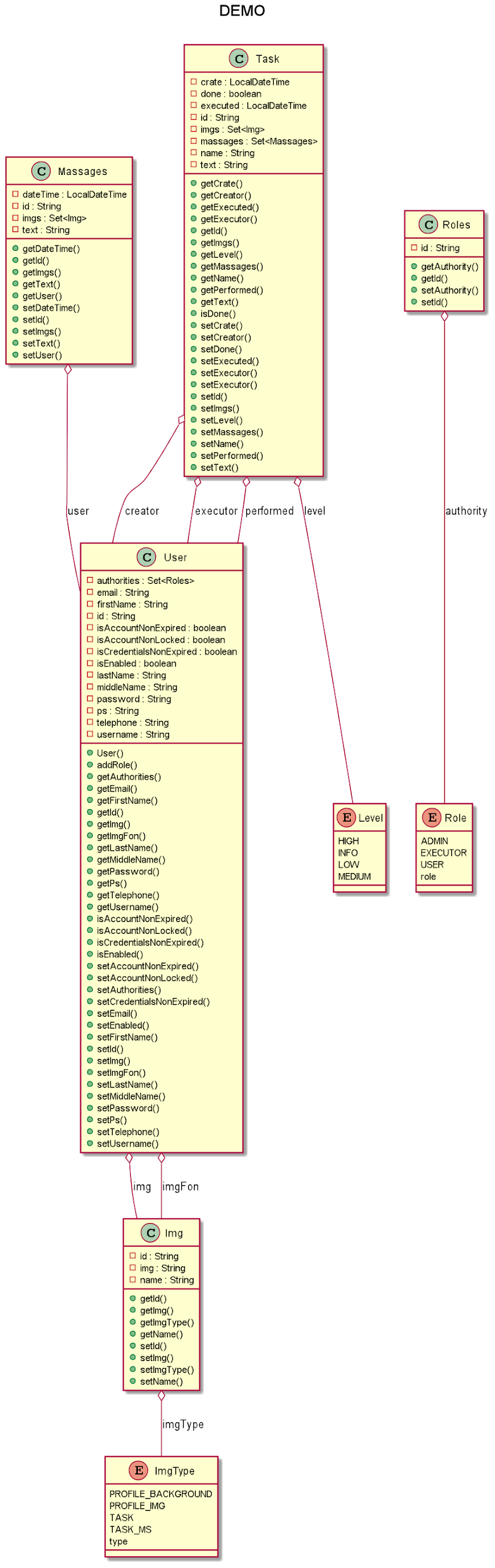
Рисунок 1 - UML class diagram Demo

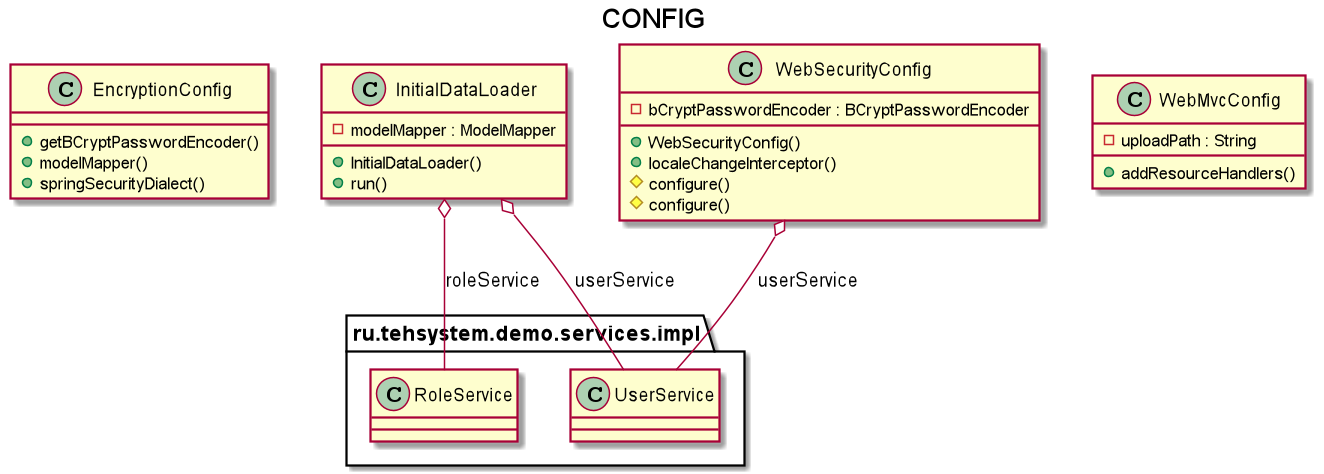
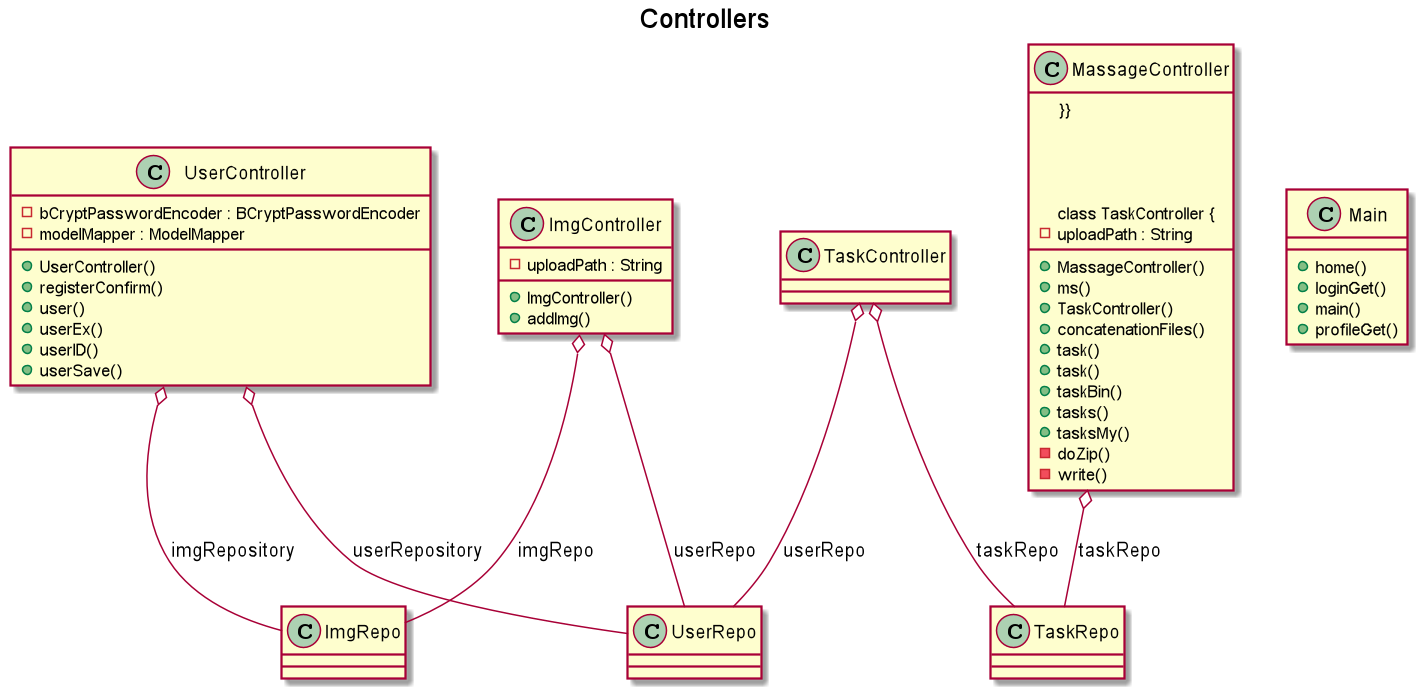
Диаграмма классов Config представлена на рисунке 2. Классы подгружаются фреймворком Spring Boot. В классе EncryptionConfig идет конфигурация bean, в классе InitialDataLoader создаются данные по умолчанию в БД, WebMvcConfig настраивает доступ к пакетам загрузки,WebSecurityConfig настройка доступа.

Рисунок 2 - UML class diagram Config

Диаграмма классов Controllers представляет из себя основные классы по принятия запросов и отправки данных клиенту.В классе UserController идет работа с пользователем,в классе TaskController работа с заданиями ,в MassageController работа с сообщениями Main отправляет страницы формата html пользователю,ImgController обрабатывает изображения.

Рисунок 3 - UML class diagram Controllers

Полностью детализированная диаграмма представлена по ссылке

* 1. **Тестирование программного продукта**

Был испльзован фреймворк для модульного тестирования Java программ, принадлежащий семейству фреймворков xUnit для различных языков программирования (например: CPPUnit – C++, JSUnit – JavaScript, NUnit — C#, PHPUnit — PHP). Основная идея модульного тестирования заключается в проверке корректности работы отдельных модулей исходного кода программы, в условиях изолированности тестируемого модуля от других. JUnit играет важнейшую роль в технике разработки через тестирование (test-driven development – TDD), которая заключается в повторении очень коротких циклов разработки: сначала пишется тест, покрывающий желаемое изменение, и только затем пишется код, который позволит пройти этот тест.

**набор тестов**

Имеются n видов тестов

1. Работа с БД
   1. Заполнение таблиц
   2. Редактирование таблиц
   3. Удаление таблиц
   4. Ввод на некорректное значение
2. Безопасность
   1. Проверка прав доступа
   2. Проверка ограничение доступа
   3. Проверка на авторизованность
3. Ввод вывод
   1. были проверены на ограничение вывода данных
   2. контроль получения данных
   3. проверка на правильность полученных данных
   4. проверка на доступ к данным

В ходе тестирования были выявлены минимальные отклонения, которые закрываются визуальной или логической безопасностью сервера.

Для тестирования был выбран метод черного ящика, при помощи знакомым были высланы ссылки на beta версию сайта которая работала 4ч 15 с , в это время пользователям были выбраны 3 роли в администратор ( имеющий доступ ко всему), исполнитель ( способен выполнять задания или создавать и выполнять задания в случае если это задание было адресовано всем или лично ему) и пользователь ( может только создавать запросы ) приведены на рисунках 4,5,6.

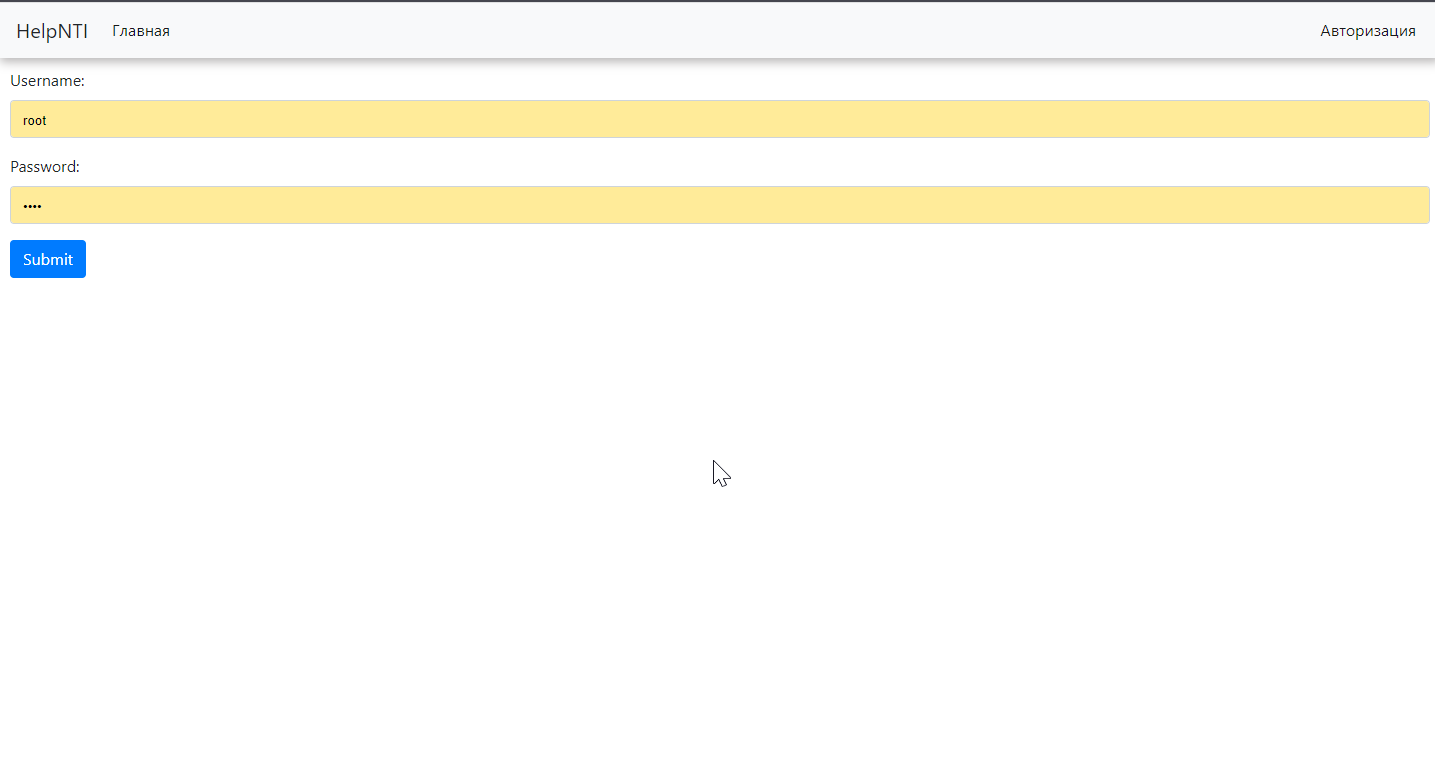


Рисунок 4 - Ввход

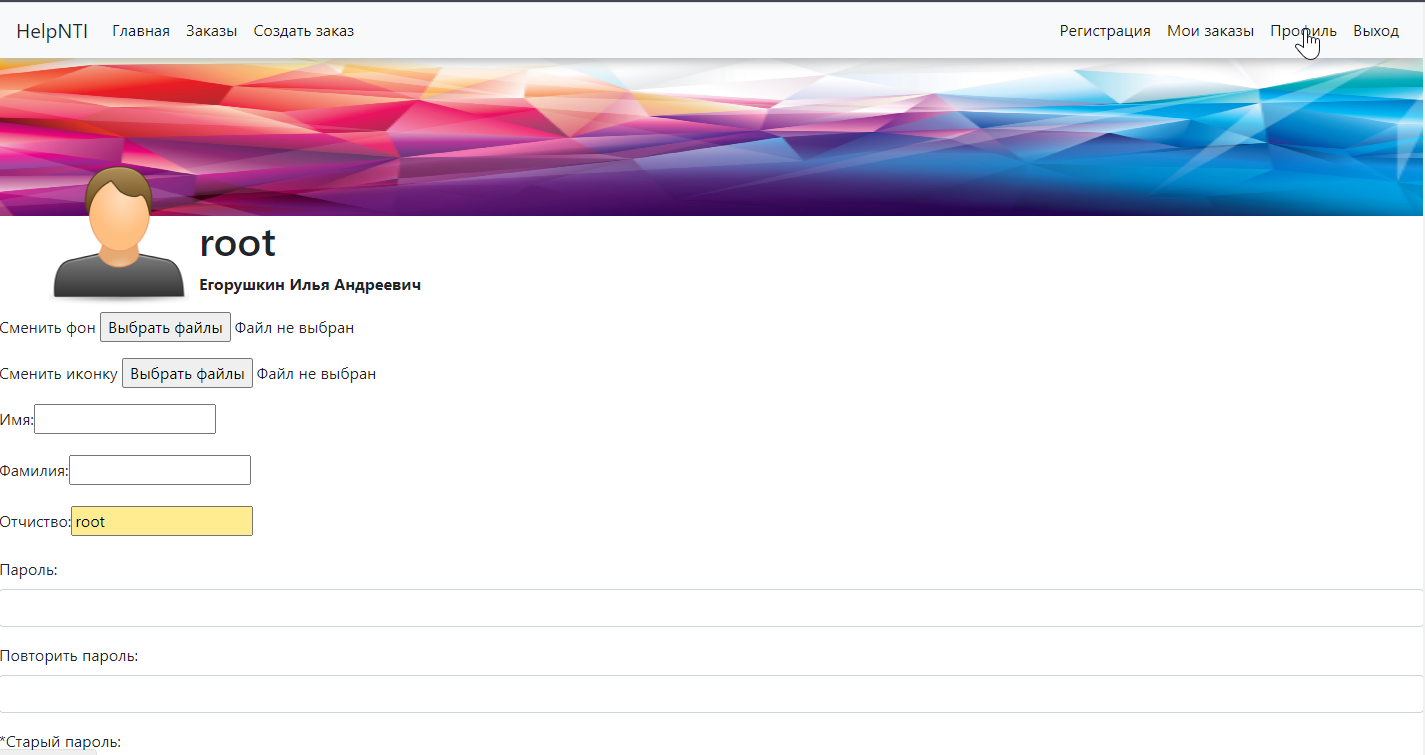


Рисунок 5 – Окно пользователя



Рисунок 6 – Создание задания

1. **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**
2. JUnit / Тестирование в Java [Электронный ресурс] – Режим доступа: https://javarush.ru/groups/posts/605-junit (дата обращения 26.01.2021)
3. JUnit 5 [Электронный ресурс] – Режим доступа: https://junit.org/junit5/ (дата обращения 26.01.2021)
4. Тестирование программы, JUnit [Электронный ресурс] – Режим доступа: http://java-online.ru/blog-junit.xhtml (дата обращения 26.01.2021)
5. Core Technologies [Электронный ресурс] – Режим доступа: https://docs.spring.io/spring-framework/docs/current/reference/html/core.html#spring-core (дата обращения 26.01.2021)
6. Testing [Электронный ресурс] – Режим доступа: https://docs.spring.io/spring-framework/docs/current/reference/html/testing.html#testing (дата обращения 26.01.2021)
7. Data Access [Электронный ресурс] – Режим доступа: https://docs.spring.io/spring-framework/docs/current/reference/html/data-access.html#spring-data-tier (дата обращения 26.01.2021)
8. OpenAPI 3 Library for spring-boot [Электронный ресурс] – Режим доступа: https://springdoc.org/ (дата обращения 26.01.2021)
9. Урок 4: Конвертация типов, форматирование значений и валидация полей | Spring по-русски! [Электронный ресурс] – Режим доступа: https://spring-rojects.ru/guides/lessons/lesson-4/ (дата обращения 26.01.2021)
10. ГОСТ 19.201-78, скачать ГОСТ 19.201-78 [Электронный ресурс] – Режим доступа: https://gostexpert.ru/gost/gost-19.201-78 (дата обращения 26.01.2021)
11. ГОСТ 19.404-79, скачать ГОСТ 19.404-79 [Электронный ресурс] – Режим доступа: https://gostexpert.ru/gost/gost-19.404-79 (дата обращения 26.01.2021)
12. ГОСТ 19.505-79, скачать ГОСТ 19.505-79 [Электронный ресурс] – Режим доступа: https://gostexpert.ru/gost/gost-19.505-79 (дата обращения 26.01.2021)
13. ГОСТ 19.301-79, скачать ГОСТ 19.301-79 [Электронный ресурс] – Режим доступа: https://gostexpert.ru/gost/gost-19.301-79 (дата обращения 26.01.2021)
14. ГОСТ 19.401-78, скачать ГОСТ 19.401-78 [Электронный ресурс] – Режим доступа: https://gostexpert.ru/gost/gost-19.401-78 (дата обращения 26.01.2021)

# ПРИЛОЖЕНИЯ

# ПРИЛОЖЕНИЕ А

(обязательное)

# Техническое задание

# 1 Ведение

Название программы: «OI System server»

программа, позволяющая предоставление данных с сервера , а также вносить или добавлять задачи в БД.

Задача сервера предоставлять:

доступ к ресурсам БД

доступ к различным файлам расположенных на сервере

* создание заказов
* удаление заказов
* отмечать выполнение заказы

# 2 Основания для разработки

# 3 Назначение разработки

**3.1 Функциональное назначение**

# Программа предоставляет возможность структурирование рабочего процесса для персонала «Отдела информатизации».

**3.2 Эксплуатационное назначение**

Программа является компонентом для структурирования и быстрой выдачи информации с сервера.

# 4 Требования к программе

**4.1 требования к функциональным характеристикам**

Программа состоит из двух основных компонент: клиентской и серверной частей, между которыми должно быть налажено взаимодействие

На серверной части должен быть реализован алгоритмы создание, редактирование, удаление заказов, создание пользователей с разными правами.

Взаимодействие между клиентской и серверной частями должно осуществляться посредством HTTP-запросов. При получении GET-запроса от клиента, сервер должен ответить сообщением в формате JSON.

**4.2 Требования к надежности**

Пользователю, работающему с программой через веб-браузер должен быть предоставлен непрерывный доступ к веб-приложению, расположенному по определённому url-адресу. Веб-сервис не должен непредвиденно прерывать свою работу.

В случае отказа работы серверной части и последующей недоступности веб-приложения, время восстановления не должно превышать одни рабочие сутки.

После запуска программы на сервере отказ программы вследствие некорректных действий оператора должен быть исключён. В том числе должна быть исключена возможность непреднамеренного выключения программы, не связанного с техническими неполадками сервераусловия эксплуатации;

**4.3 Условия эксплуатации**

Требований к климатическим условиям эксплуатации не предъявляется

**4.3.2. Требования к видам обслуживания**

Обслуживание не требуется.

**4.3.3. Требования к численности и квалификации персонала**

Для управления системой достаточно одного человека, способного запустить на сервере систему управления базами.

Требуемая квалификация пользователя - оператор ЭВМ

**4.4 Требуемая квалификация и уровень подготовки пользователя**

Для работы с программой требуются человек, имеющий область представления настроек серверов Linux.

**4.5 Требования к составу и параметрам технических средств**

* Серверная часть должна быть написана на языке программирования Java 8. БД должна быть PostgreSQL и должна работать на Linux серверах опционально должна быть возможность запуска на Windows Server.Сервер – 8Гб ОЗУ, Процессор 8 ядра.

**4.6 Требования к информационной и программной совместимости**

Программа поставляется в виде программного jar-файла изделия на внешнем носителе информации – флэшке, на котором содержится приложение.

**4.7 Требования к маркировке и упаковке**

Программа поставляется в виде jar-файла.

Программа поставляется в виде программного изделия на внешнем носителе - USB-флеш накопитель. Программное изделие должно иметь маркировку с обозначением наименования изделия

**4.9 Специальные требования**

Серверная часть должна быть написана на языке программирования Java 8. БД должна быть PostgreSQL и должна работать на Linux серверах опционально должна быть возможность запуска на Windows Server.

# 5 Требования к программной документации

# Техническое задание (ГОСТ 19.201-78);

# Пояснительная записка (ГОСТ 19.404-79);

# Руководство оператора (ГОСТ 19.505-79);

# Программа и методика испытаний (ГОСТ 19.301-79);

# Текст программы. (ГОСТ 19.401-78);

Документы к программе должны быть выполнены в соответствии с ГОСТ 19.106-78 и ГОСТами к каждому виду документа (см. п. 5.1.); Документация и программа также сдается в электронном виде в формате .pdf или .docx. в архиве формата .zip или .rar;

# 6 Технико-экономические показатели

# В рамках данной работы расчёт экономической эффективности не предусмотрен. Использование разрабатываемого инструмента сократит врем, затрачиваемое на обращение в « Отдел информатизации».7 Стадии и этапы разработки

Стадии и этапы разработки отображены в таблице 1.

Таблица 1 – стадии и этапы разработки

|  |  |  |
| --- | --- | --- |
| № | Этапы разработки | Сроки |
| 1 | Техническое задание |  |
| 2 | Эскизный проект |  |
| 3 | Технический проект |  |
| 4 | Рабочий проект |  |
| 5 | Внедрение |  |

# Исполнитель: Егорушкин Илья Андреевич.

# 8 Порядок контроля и приемки

Контроль и приемка разработки осуществляются в соответствии с документом «Программа и методика испытаний».

Производится проверка корректного выполнения программой заложенных в нее функций, то есть осуществляется функциональное тестирование программы. Функциональное тестирование осуществляется в соответствии с документом «Программа методика испытаний».

Прием программы будет утвержден при корректной работе программы.