Студенты ФИТ НГУ

Котов Родион

Курдюков Данил

группа 20208

Версия 1.0.0

Содержание

[1. Введение 3](#_Toc122283001)

[1.1 Цель 3](#_Toc122283002)

[1.2 Область действия 3](#_Toc122283003)

[1.3 Определения и сокращения 3](#_Toc122283004)

[1.4 Ссылки 3](#_Toc122283005)

[1.5 Краткое описание 3](#_Toc122283006)

[2. Предметная область проекта 4](#_Toc122283007)

[2.1 Существующие проблемы 4](#_Toc122283008)

[2.2 Предполагаемое решение 4](#_Toc122283009)

[3. Требования к программному решению 4](#_Toc122283010)

[3.1 Роли 4](#_Toc122283011)

[3.2 Функциональные требования для роли Не авторизированный пользователь 4](#_Toc122283012)

[3.2.1 Регистрация 4](#_Toc122283013)

[3.2.2 Авторизация 5](#_Toc122283014)

[3.3 Функциональные требования для роли Авторизированный пользователь 5](#_Toc122283015)

[3.3.1 Выход 5](#_Toc122283016)

[3.4 Функциональные требования для роли Сотрудник 5](#_Toc122283017)

[3.4.1 Просмотр любых заказов 5](#_Toc122283018)

[3.4.1.1 Скачивать файл 5](#_Toc122283019)

[3.4.2 Редактирование любых заказов 5](#_Toc122283020)

[3.4.2.1 Изменить срок хранения 5](#_Toc122283021)

[3.4.2.2 Изменить статус 5](#_Toc122283022)

[3.5 Функциональные требования для роли Клиент 6](#_Toc122283023)

[3.5.1 Просмотр своих заказов 6](#_Toc122283024)

[3.5.1.1 Скачивание файлов 6](#_Toc122283025)

[3.5.1.2 Удаление своих заказов 6](#_Toc122283026)

[3.5.2 Создание заказа 6](#_Toc122283027)

[3.6 Нефункциональные требования 7](#_Toc122283028)

[4. Обзор архитектуры 7](#_Toc122283029)

[4.1.1 Компонентная модель системы 7](#_Toc122283030)

[4.1.1.1 Компонент 1 8](#_Toc122283031)

[4.1.1.2 Компонент 2 8](#_Toc122283032)

[4.1.2 Компоненты сторонних производителей 8](#_Toc122283033)

[4.1.3 Схема развертывания приложения 8](#_Toc122283034)

[5. Допущения и ограничения 8](#_Toc122283035)

[6. Известные проблемы 9](#_Toc122283036)

[6.1 Невысокая производительность приложения 9](#_Toc122283037)

# Введение

## Цель

Данный документ представляет собой техническое описание проекта *FileDrop* и содержит основные требования к разрабатываемой в рамках проекта программной системе и описание архитектуры программного решения.

## Область действия

Документ разработан в рамках проекта FileDrop на основе стандартного шаблона и предназначен для использования студентами ФИТ и преподавателями дисциплины ООАД.

## Определения и сокращения

Таблица 1: Определения и сокращения

|  |  |
| --- | --- |
| Термин | Описание |
| API | Application Programming Interface — описание способов взаимодействия одной компьютерной программы с другими. |

## Ссылки

В тексте содержатся ссылки на следующие документы:

1. <Имя файла документа>, v<версия> - <описание документа>

Ссылки приводятся в виде [N], где N – номер документа в вышеприведенном списке.

## Краткое описание

Программное обеспечение представляющие собой набор API для передачи файлов неограниченного размера между пользователями сети с фиксированным временем хранения.

# Предметная область проекта

При оказании копировальных услуг, возникает потребность в передаче данных между сотрудником и клиентом организации.

## Существующие проблемы

|  |  |
| --- | --- |
| Проблемы | Решения |
| Различные способы передачи данных, которые накладывают различные ограничения, связанные с размером, типом передаваемых данных | Создание единой точки сбора данных с собственными ограничениями, необходимых для нужд организации |
| Накопление большого количества данных, в следствие оказания услуг | Организация системы хранения и контроля данных |

## Предполагаемое решение

Создание сетевого приложения для локальной машины организации, осуществляющее организацию и контроль единого места передачи и хранения данных.

# Требования к программному решению

Данный раздел описывает требования к программной системе, разрабатываемой в рамках проекта FileDrop.

## Роли

Роль — это что-то (например: другая система) или кто-то (например: человек) вне системы, которые взаимодействуют с ней. В предлагаемой к разработке системе идентифицированы следующие роли:

1. Пользователь – человек, использующий веб-приложение
2. Авторизированный пользователь – пользователь, вошедший в систему
3. Не авторизированный пользователь – пользователь, не вошедший в систему
4. Сотрудник – авторизированный пользователь, являющийся представителем организации
5. Клиент - авторизированный пользователь, пользующийся услугами организации

## Функциональные требования для роли Не авторизированный пользователь

### Регистрация

Триггер – нажатие кнопки “РЕГИСТРАЦИЯ” в header(“шапка”) веб-приложения

Основной сценарий:

1. Указание имени
2. Указание персонально номера
3. Соглашение с обработкой данных организацией, нажатием на checkbox
4. Нажатие кнопки “ВОЙТИ”, с последующим созданием пользователя в системе
5. Открытие страницы личного рабочего места пользователя

### Авторизация

Триггер – нажатие кнопки “ВОЙТИ” на главной странице веб-приложения

Основной сценарий:

1. Указание персонально номера
2. Указание пароля
3. Нажатие кнопки “ВОЙТИ”, с последующим созданием пользователя в системе

## Функциональные требования для роли Авторизированный пользователь

### Выход

Триггер – нажатие кнопки “ВЫЙТИ”

Основной сценарий:

1. Нажатие кнопки “ ВЫЙТИ”, с последующим выходом и переадресацией на главную страницу

## Функциональные требования для роли Сотрудник

### Просмотр любых заказов

Основной сценарий:

1. Нажать на поле ввода “НОМЕРА ЗАКАЗА”
2. Ввести номер заказа
3. Нажать на кнопку “Поиск”

#### Скачивать файл

Триггер – нажатие кнопки “Скачать”

Основной сценарий:

1. Скачать файл, нажав кнопку “Скачать”

### Редактирование любых заказов

Основной сценарий:

1. Нажать на поле ввода “НОМЕРА ЗАКАЗА”
2. Ввести номер заказа
3. Нажать на кнопку “Поиск”

#### Изменить срок хранения

Триггер – нажатие кнопки “Изменить”

Основной сценарий:

1. Выбрать дату хранения
2. Сохранить изменения, нажав кнопку “Сохранить”

#### Изменить статус

Триггер – нажатие кнопки “Выполнено”

Основной сценарий:

1. Изменение статуса заказа на “Выполнено”

## Функциональные требования для роли Клиент

### Просмотр своих заказов

Триггер – нажатие на любой заказ из списка “Мои заказы”

Основной сценарий:

1. Открытие окна для просмотра заказа
2. Переход назад кнопкой “←”

#### Скачивание файлов

Триггер – нажатие кнопки “Скачать” у нужного файла заказа

Основной сценарий:

1. Открытие окна, предлагающего сохранить файл в нужный каталог
2. Скачивание файла, нажав кнопку “Скачать”

### 3.5.1.2 Удаление своих заказов

Триггер – нажатие на кнопку “Удалить”

Основной сценарий:

1. Удаление заказа

### Создание заказа

Триггер – нажатие кнопки “Добавить заказ”

Основной сценарий:

1. Открытие окна для создания заказа
2. Заполнение полей
3. Отправка заказа на исполнение, нажатием кнопки “Создать”

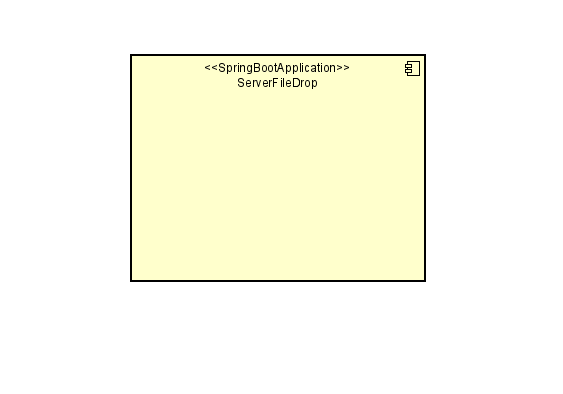
## Нефункциональные требования

* Совместимость со популярными браузерами последних версий.
* Возможность установки системы на любую машину
* Необходимость наименьшего зависимости от ресурсов ПК

# Обзор архитектуры

Этот раздел описывает архитектуру системы.

### Компонентная модель системы



#### ServerFileDrop

### Схема развертывания приложения

#### Client

Клинтом является любой интерфейс поддерживающий просмотр страниц в HTML и взаимодействие с сервером по протоколу HTTP

#### AppServer

Сервером является любой компьютер организации, с предустановленным пакетом JDK(версией не старше 17).

##### ServerFileDrop

Spring Boot приложение, развертуемое на машине сервера с помощью пакета JDK

##### БД H2

In-memory база данных, автоматически создаваемое на машине сервера в каталоге, а котором запущен ServerFileDrop

# Допущения и ограничения

[Краткое описание допущений, которые подразумевает данный проект, и любых ограничений (например, по бюджету, участникам, требуемому оборудованию, срокам и т.п.), накладываемых на его выполнение.]

Пример: При разработке проекта принято допущение, что число транзакций в единицу времени значительно (более чем в 10 раз) снижается в ночное время, что позволяет в период с 01:00 до 6:00 производить автоматическое обновление программного обеспечения системы, требующее полной перезагрузки и остановки сервиса на период до 5 минут.

# Известные проблемы

Ниже приводятся известные на данный момент проблемы и недоработки выработанного программного решения, а также возможные пути их устранения в последующих итерациях проекта.

## Невысокая производительность приложения

|  |  |
| --- | --- |
| Проблема | Производительность приложения экспоненциально деградирует при общем числе пользователей выше 10000 и числе одновременных сессий выше 100. |
| Ранг | 10 (высокий) |
| Влияние на проект | Невозможность использования системы при числе пользователей более 10000. |
| Пути решения | Кластеризация веб-сервера и сервера базы данных, а также применение load balancer в точке маршрутизации запроса к веб-серверу. |

Лист регистрации изменений

|  |  |  |  |
| --- | --- | --- | --- |
| Дата | Версия | Описание | Автор |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

[В качестве описания версии можно указывать какие изменения/дополнения были сделаны в этой версии по отношению к предыдущей.]

Лист регистрации проверок

|  |  |  |  |
| --- | --- | --- | --- |
| Дата | Версия | Описание | Автор |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

[Здесь описываются результаты проверки документа. Для каждой проверки указывается число, версия документа, описание результатов проверки и имя человека, который делал проверку.]