FileDrop

Студенты ФИТ НГУ

Котов Родион

Курдюков Данил

группа 20208

Версия 1.0.0

Содержание

[1. Введение 3](#_Toc211722981)

[1.1 Цель 3](#_Toc211722982)

[1.2 Область действия 3](#_Toc211722983)

[1.3 Определения и сокращения 3](#_Toc211722984)

[1.4 Ссылки 3](#_Toc211722985)

[1.5 Краткое описание 3](#_Toc211722986)

[2. Предметная область проекта 4](#_Toc211722987)

[2.1 Существующие проблемы 4](#_Toc211722988)

[2.2 Предполагаемое решение 4](#_Toc211722989)

[3. Требования к программному решению 4](#_Toc211722990)

[3.1 Роли 4](#_Toc211722991)

[3.2 Функциональные требования для роли Роль1 4](#_Toc211722992)

[3.2.1 <Use Case Name 1> 4](#_Toc211722993)

[3.2.2 <Use Case Name 2> 4](#_Toc211722994)

[3.3 Функциональные требования для роли Роль2 4](#_Toc211722995)

[3.3.1 <Use Case Name 1> 4](#_Toc211722996)

[3.3.2 <Use Case Name 2> 4](#_Toc211722997)

[3.4 Нефункциональные требования 4](#_Toc211722998)

[4. Обзор архитектуры 5](#_Toc211722999)

[4.1.1 Компонентная модель системы 5](#_Toc211723000)

[4.1.1.1 Компонент 1 5](#_Toc211723001)

[4.1.1.2 Компонент 2 5](#_Toc211723002)

[4.1.2 Компоненты сторонних производителей 5](#_Toc211723003)

[4.1.3 Схема развертывания приложения 5](#_Toc211723004)

[5. Допущения и ограничения 5](#_Toc211723005)

[6. Известные проблемы 6](#_Toc211723006)

[6.1 Невысокая производительность приложения 6](#_Toc211723007)

# 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 : 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:

<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

|  |  |  |
| --- | --- | --- |
| Проблема | Влияние | Способ решения |
| Ограничение на объем предаваемого файла, накладываемое средствами обмена данными | Ухудшение качества сервиса, вызванного поиском альтернативных путей | Написание унифицированного средства для передачи файлов между клиентом и организацией |
| Накопление большого количества данных, в следствие оказания услуг | Накладные расходы, связанные с организацией хранения файлов | Создание механизма контроля срока хранения файлов |

## Предполагаемое решение

Создание сетевого приложения для локальной машины организации, которое будет позволять:

1. Клиенту и организации, обмениваться файлами
2. Организации определять период хранения файлов
3. Сотруднику организации корректировать время хранения файлов

# Требования к программному решению

Данный раздел описывает требования к программной системе, разрабатываемой в рамках проекта <Название проекта>.

## Роли

*[Как не сложно заметить, роль – это actor в терминах UML]*

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. <Роль1> – <краткое описание роли>
2. <Роль2> – <краткое описание роли>

## Функциональные требования для роли Роль1

сссссс сделать описание требований к системе в соответствии с Use-Case моделью. Для каждой роли необходимо ввести отдельный пункт 2-го уровня, такой как 3.2]

### <Use Case Name 1>

[В этом пункте необходимо сделать описание данного Use-Case.]

### <Use Case Name 2>

[В этом пункте необходимо сделать описание данного Use-Case.]

## Функциональные требования для роли Роль2

### <Use Case Name 1>

[В этом пункте необходимо сделать описание данного Use-Case.]

### <Use Case Name 2>

[В этом пункте необходимо сделать описание данного Use-Case.]

## Нефункциональные требования

[В этом пункте необходимо описать нефункциональные требования, такие как:

* Производительность
* Масштабируемость
* Ограничения по используемым компонентам
* Необходимость миграции данных из legacy систем
* И т.д.]

# Обзор архитектуры

Этот раздел описывает архитектуру системы.

### Компонентная модель системы

*[Здесь приводится Component diagram - диаграмма компонентов системы, со связями между компонентами и интерфейсами между ними, а также описание их взаимодействия. Для каждого компонента дается краткое описание его места и предназначения в системе]*

#### Компонент 1

*[Здесь приводится более подробное описание предназначения компонента и Package diagram – диаграмма пакетов, из которых состоит данный компонент. Обязательно выделение на диаграмме интерфейсов пакета, служащих для связи с другими пакетами (фасад пакета), а также ключевых классов, используемых другими пакетами в use-case реализациях]*

#### Компонент 2

*[Здесь приводится более подробное описание предназначения компонента и Package diagram – диаграмма пакетов, из которых состоит данный компонент. Обязательно выделение на диаграмме интерфейсов пакета, служащих для связи с другими пакетами (фасад пакета), а также ключевых классов, используемых другими пакетами в use-case реализациях]*

### Компоненты сторонних производителей

*[Здесь приводится список использованных компонент сторонних производителей, использованных при разработке системы, с указанием их предназначения в системе]*

### Схема развертывания приложения

*[Здесь приводится Deployment diagram - диаграмма развертывания системы, со связями между узлами и указанием способа связи (протокола). На диаграмме обязательно указать, какие компоненты находятся на том или ином узле]*

# 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.]