|  |  |  |  |
| --- | --- | --- | --- |
| МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ  ФГБОУ ВО «Пермский государственный национальный  исследовательский университет» | | | |
|  | | | |
| ОТЧЕТ  по лабораторной работе №3  «Разработка распределенного приложения, использующего технологию веб-служб»  по дисциплине «Технологии разработки распределенных приложений» | | | |
| Работу выполнили студенты группы ФИТ-4-НБ 4 курса:  Ботова Карина  «\_\_\_\_» \_\_\_\_\_\_\_\_\_ 2020 г. |  | | Проверил к.ф.-м.н., доцент кафедры ПМИ  Деменев Алексей Геннадьевич  «\_\_\_\_» \_\_\_\_\_\_\_\_\_ 2020 г. |
|  | |  | |
| Пермь 2020 | | | |

1. Постановка задачи

***Цель***: изучение возможностей технологии веб-служб для создания распределенных приложений.

***Формируемая компетенция:*** способность применять на практике теоретические основы и общие принципы разработки распределенных систем; способность использовать на практике стандарты сетевого взаимодействия компонент распределенной системы.

***Организация выполнения работы:*** Каждый студент выполняет индивидуальное задание.

***Требования к выполнению работы:***

Необходимо реализовать веб-сервис и приложение, его использующее

1. Веб-сервис и приложение, его использующее, должны быть разработаны на разных объектно-ориентированных языках программирования.

2. Веб-сервис должен принимать параметры и передавать приложению результат.

3. В качестве параметров должны передаваться объекты классов, написанных самостоятельно, т.е. не должны передаваться строки, числа или другие простейшие типы.

По окончании выполнения задания каждый студент должен подготовить отчет.

Отчет по выполнению задания должен включать:

1. Общее описание приложения. Постановка задачи, введение в предметную область.
2. Архитектура системы. Обоснование выбора данного типа архитектуры распределенного приложения. Алгоритм работы приложения в целом.
3. Архитектура каждого из логических компонент системы (серверы, клиенты). Подходы к реализации. Алгоритмы работы.
4. Методы коммуникаций компонентов системы (клиент→сервер, сервер→клиент и т.д.).
5. Структура передаваемых данных.
6. Исходный код.

Распределение баллов за выполнение работы представлено в следующей таблице:

|  |  |
| --- | --- |
| **Критерий оценивания** | **Оценка** |
| Приложение, написанное студентом, работает в сети Интернет без сбоев. | 2 |
| Веб-сервис и приложение, его использующее, написаны на разных языках программирования. | 1 |
| Веб-сервис принимает от приложения исходные данные и возвращает результат. | 1 |
| В качестве параметров веб-сервис принимает объекты классов. | 2 |

1. Описание приложения. Введение в предметную область

Предметная область – генерирование пароля.

Пользователь задает длину желаемого пароля. Сервер генерирует пароль и отправляет его клиенту.

1. Лицензия

Приложение разработано на языке программирования JavaScript и распространяется с лицензией MIT License (URL: https://mit-license.org/ (дата обращения: 29.06.20)). Сервер написан на языке программирования Go и распространяется с лицензией MIT License (URL: https://mit-license.org/ (дата обращения: 29.06.20)), Creative Commons Attribution 3.0 License (URL: https://creativecommons.org/licenses/by/3.0/ (дата обращения: 29.06.20)), BSD license (URL: https://golang.org/LICENSE (дата обращения: 29.06.20)).

1. Архитектура системы

На рисунке 1 изображена архитектура системы.

User

Client

Server

Рисунок 1 – Архитектура системы

**Алгоритм работы системы следующий:**

1. Пользователь запускает программу клиента и передает через аргумент командной строки желаемую длину пароля.
2. Клиент обращается к серверу по протоколу gRPC, передавая в запросе объект класса PasswordGeneratorReq, в полях которого задана желаемая длина пароля.
3. Сервер принимает запрос, генерирует пароль исходя из заданной длины и возвращает объект класса PasswordGeneratorRes, в поле которого содержится сгенерированный пароль.
4. Архитектура клиента

Клиент представляет собой консольное приложение, написанное на языке JavaScript. Архитектура клиента изображена на рисунке 2.

Задачи клиента – передать длину желаемого пароля на сервер и отобразить результат пользователю.

Client

User

Server

Коммуникационный модуль

Рисунок 2 – Архитектура клиента

6 Архитектура сервера

Сервер представляет собой консольное приложение, архитектура сервера изображена на рисунке 3.

Задача сервера: принять длину желаемого пароля и сгенерировать случайный пароль.

Client

Server

Рисунок 3 – Архитектура сервера

1. Методы коммуникаций компонентов системы

Для все коммуникации в системе используется синхронное взаимодействие по протоколу gRPC. Передаваемый формат данных – protobuf3

1. Структура передаваемых данных

message PasswordGeneratorReq {

int32 length = 1;

}

message PasswordGeneratorRes {

string password = 1;

}

9 Исходный код

Реализация приложена рядом с отчетом.

Проект - <https://github.com/KarinaBotova/Password>