

**Computer Science Department**

**CCP6427: Cloud Engineering**

**Spring 2022**

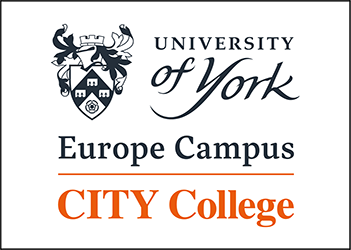
**Design and development of a microservices-based application**

**Submission Deadline: 22/5/2022**

**Actual Submission Date: 22/5/2022**

|  |
| --- |
| **Student Registration Number/s** |
| WMY21040 |

**Declaration**: *I have completed and submitted this work by myself without assistance from or communication with another person either external or fellow student. I understand that not working on my own will be considered grounds for unfair means and will result in fail mark for this work and might invoke disciplinary actions. It is at the instructor’s discretion to conduct an oral examination, if necessary, which will result in the award of the final grade for that particular piece of work.*



Report

**File Conversion Microservices with Eureka**

**Abstract (250 – 300 words)**

My abstract here.

**Keywords:** Microservices, Springboot, Eureka Netflix, Model View Controller (MVC), Representational State Transfer (REST).

Table of Contents

[1. Introduction 1](#_Toc89161934)

[2. First Section 2](#_Toc89161935)

[2.1 First Sub-section 2](#_Toc89161936)

[3. Conclusion 3](#_Toc89161937)

# 

**Table of Figures**

[Figure 8‑1. A figure A-1](#_Toc89161938)

**Table of Tables**

[Table 1‑1. Hello 1](#_Toc89161939)

# Introduction

In this project the microservices software engineering architecture will be put in practice through a minimal system. The system in question will be responsible for receiving plain text data, converting it to a pdf file and sending it back to the client application. The goal is to create a realistic and flexible microservice application which addresses traditional issues from service-oriented design.

# System Specification

The system does not require extensive business logic, but several architectural constraints that require consideration. System requirements will be presented organised by type below:

**Architectural system requirements:**

* Microservices are designed

**Functional system requirements:**

* Users provide plain text through a client interface.
* Users expect a pdf file to be generated containing the plain text provided.
* Developers and other stakeholders can identify that the microservices are operational without observing the running instances consoles directly.

**Non-Functional system requirements:**

* A user-friendly client interface is created that presents the actions the user can take with the system. Also, any limitations with the service’s functionality needs to be transparently communicated to users in the user interface.
* Feedback should be provided to users in response to their actions for user experience purposes.

# Design

# Implementation

# Testing

# Conclusions

A Figure

Figure 3‑1. A figure

Figure 2