**Cork Institute of Technology**

**Design Document**

**Cloud Development Frameworks Assignment 2**

**Jimmy Collins – R00145569**

Contents

[Introduction 2](#_Toc481428999)

[Configuration Management 2](#_Toc481429000)

[Microservices Overview 2](#_Toc481429001)

[Cart Microservice 2](#_Toc481429002)

[Catalogue Microservice 2](#_Toc481429003)

[Front-end Microservice 2](#_Toc481429004)

[Purpose 2](#_Toc481429005)

[Codebase Notes 2](#_Toc481429006)

[How Different Things Work - (name this better) 2](#_Toc481429007)

[Users Microservice 3](#_Toc481429008)

[Stock Administration Microservice 3](#_Toc481429009)

[Order Microservice 3](#_Toc481429010)

[Microservices Communication 3](#_Toc481429011)

# Introduction

TODO

# Configuration Management

TODO

# Microservices Overview

TODO

## Cart Microservice

TODO

## Catalogue Microservice

TODO

## Front-end Microservice

### Purpose

The front-end service is responsible for defining and managing the main user interface.

### Codebase Notes

The api folder contains wrappers that enable the front-end to talk to the other services. The endpoint URL’s of the services are contained in endpoints.js.

NOTE – will need to add wrapper for stock management service here.

NOTE – looks like the orders service does not exist – will I need to add this to implement the checkout feature?

### How Different Things Work - (name this better)

#### Logging In

When a user logs in the following sequence of events happens (triggered via jQuery):

* The submit of the login form is detected.
* The code prevents moving to a new page (check this – event.preventDefault());
* A new variable is created that is created from the values in the login form (the username and password).
* This is then turned into JSON.
* Next this is posted to /login (contained in the api/users/index.js – the ‘helper’ for the users service).
* This login then posts the JSON body to the users service and checks the response code.
* If the response code is HTTP 200 the body is parsed and returned.
* If an error occurred, the status of the request is set to HTTP 500 and the error output to the console.
* A cookie is also created on the client.

## Users Microservice

TODO

## Stock Administration Microservice

TODO

## Order Microservice

TODO

# Microservices Communication

TODO