



# **Assessment Project**

**Microservices Engineer Assessment Project v1.0**

## TABLE OF CONTENTS

<b>1 Background</b>	<b>2</b>
1.1 Approach	2
1.2 Assessment	2
1.3 Standards to follow	2
<b>2 Solution Requirements</b>	<b>3</b>
2.1 Background	3
2.2 The coffee shop app	3
2.3 The customer app	3
<b>3 Deliverables</b>	<b>5</b>
3.1 Part 1	5
3.2 Part 2	5

# 1 Background

This document outlines a mini project that is used to assess platform engineers at 101 Digital.

## 1.1 Approach

Candidates taking this assessment should read and understand the following items clearly:

1. The mini project requirements outlined in this document
2. The assessment criteria outlined below
3. The deliverables to be sent to 101 Digital

## 1.2 Assessment

You will be assessed on the following assessment criteria:

1. **Packaging** - how well you package your deliverables to make it easy for us to access, build, deploy, test, review
2. **Design** - clear design of the APIs and any associated items such as databases
3. **Working** - a working API / APIs demonstrated through an app / web page or test scripts
4. **Documentation** - clear documentation of design, deployment and testing
5. **Deployment** - ease of deployment and testing the code / APIs
6. Value Add - additional items that you have considered and contributed to the project (that we have not requested in the requirements)

## 1.3 Standards to follow

The following section outlines the standards you should follow:

1. Use Java Spring Boot framework to develop the APIs
2. Use Postgres as database where a database is required
3. We will be deploying and testing your services on AWS using Docker containers
4. We prefer completely automated database setup (eg. Liquibase)

## 2 Solution Requirements

The requirements for the overall solution is outlined below:

### 2.1 Background

A global coffee shop chain / franchise intends to launch an app to allow their regular customers to pre-order coffee to pick up (say, on their way to work).

They have identified the following needs:

1. The coffee shop chain is a global network. So, they need to service shop locations across multiple geographies.
2. The space is quite limited. So, they want everything to work easily on an app.
3. They need two apps (a) one for the shop owner and (b) one for the customer
4. They have decided to build (a) on Android and (b) on iOS and Android
5. Not all coffee shops have the same menu. So, they need to be able to handle a menu based on the shop
6. Most of their shops have only one queue, but some shops are able to support up to 3 queues
7. They would like their service to be API enabled, so that others (3rd parties) can build apps using their APIs

### 2.2 The coffee shop app

1. Allows the shop owner to login as an admin user
2. Allows the shop owner to setup / configure the app to support their shop
3. Allows the shop owner to configure the shop / app as follows:
  - a. Location and Contact details
  - b. The coffee menu & pricing
  - c. Number of queues and the maximum size of the queue
  - d. opening / closing times
4. Allows the shop operator to login and manage the queue
5. To view / see the size of the queue and the number of waiting customers
6. To easily view the orders placed by the customers in the queue
7. The name of the persons in the queue
8. A score indicating the number of times that customers has been served by the coffee shop chain
9. Take a customer off the queue and service them

### 2.3 The customer app

This section outlines the requirements for the customer app.

1. Allows the customer to register with their mobile number, name and regular address (home or work)
2. Allows the customer to view and find the coffee shops closest to them
3. Place an online order for a coffee from the menu
4. See their position in the queue (and expected waiting time before collecting the coffee)
5. Exit the queue at any time (and notify the shop to cancel the order)
6. Any other function that you may consider useful

## 3 Deliverables

The following section outlines what you should deliver to 101 Digital. You should only start Part 2, after Part 1 has been successfully completed and reviewed by 101 Digital.

### 3.1 Part 1

1. Create a solution design to address the requirements described above. Use use cases, concept diagrams, sequence diagrams, data designs, data flows or any other forms of diagrams needed to describe the design
2. Create a time estimate to build out the solution
3. You should plan to design and build the back-end APIs databases etc.
4. You **do NOT need to design and build any of the apps**
5. Consider / specify what coding, naming, security, technology standards you will follow when building the solution.
6. Outline the security solution that you intend to use
7. Outline the API endpoints that will be developed as part of the solution.
8. Outline how you will test the solution in the absence of the apps / front-end

### 3.2 Part 2

1. Build the solution outlined in the above Phase 1 (not web app, only APIs and test / demo scripts). Please note that you only need to **build 1 or 2 APIs - as you see fit**, to demonstrate your capability
2. Use Java Spring Boot framework to build the solution / APIs
3. Setup sample data needed to demonstrate the app
4. Use Liquibase for setting up and managing database tables
5. Using test scripts or test apps / code demonstrate the key features of the platform
6. Demonstrate how errors are detected and handled by the solution
7. Package and deploy the solution on AWS as Docker containers