Requirements Document

Introduction

This document defines all functional requirements of the project and prioritizes them using the **MoSCoW** table method. The tables are split into 2 sections – customer and business side of the application.

Functional Requirements

Customer side of application

Functionality	Must	Should	Could	Would
A customer can create an account	X			
A customer can log in their account	X			
A customer can log out of their account	X			
A customer can view all available offers	X			
A customer can purchase a bag	X			
A customer can filter offers based on location		X		
A customer can add a business to their favourites list			X	
A customer can view their account information	X			
A customer can see their history of purchases	X			
A customer can search for a business based on name		X		
A customer can view all businesses displayed on a map			X	
A customer can leave a feedback after collecting a bag from a business				X
A customer can view a business' rating				X
A customer can view their ecological impact based on purchases within the application			X	
A customer can receive achievement badges				X
A customer is awarded "points" for each purchase				X
A customer can view all available rewards which can be redeemed with "points"				X
A customer can refer other customers to the application, receiving a reward				X
A customer can add, remove or modify their payment method	X			

Business side of application

Functionality	Must	Should	Could	Would
A business owner can register their business on the platform	X			
A business owner can log in with their business account on the platform	X			
A business owner can log out of their account on the platform	X			
A business owner can create an offer with expiring goods	X			
A business owner can see the history of purchases from their store	X			
A business owner can set up a weekly schedule for offering their expiring goods		X		
A business owner can modify the amount of bags for the given day, based on availability	X			
A business owner can offer additional bags, aside from the weekly schedule on “special days”		X		
A business owner can view a calendar, which displays on which days there are bag offers scheduled			X	
A business owner can view various metrics on their store performance within the platform		X		
A business owner can receive achievement badges for their store				X
A business owner can view various metrics for their store’s ecological impact within the platform			X	
A business owner can refer another business owner to the platform for a reward			X	
A business owner can view their store’s account information	X			
A business owner can cancel an order before it is collected	X			
A business owner can receive notifications for important actions that need to be taken			X	

Non-functional requirements

Scalability

Fully developed, the application should be able to operate at both low and high loads, scaling based on the needs at the given moment. What is more, it should be able to handle high volume of data, also scaling based on current load. Different running parts of the application should be scaled based on usage independently.

- Expected requests per hour - up to 4,000,000 requests spreading across all services;

Availability

Since the project is aimed to become a publicly used web application, high availability is a key requirement for providing a good service.

- Expected uptime is ~98%;

DevOps

Every container should be deployed independently with an automated CI/CD pipeline. What is more, every container should be monitored individually.

Cloud Services

Cloud services that could be integrated within the system should be considered and utilized if they can add value to the system. In addition, all of the system's running parts should be deployed to the cloud, if possible.

Security

In order to ensure security in the finished product, some requirements have to be covered:

- **Authentication** – the system should be able to support a secure way of authenticating its users, before giving them access to protected resources;
- **Authorization** – access control has to be applied in a way which restricts authenticated users from accessing resources which are not within their given permissions;
- **OWASP coverage** – The application should be protected from OWASP's top 10 security threats, which needs to be validated by a tool, integrated in the CI pipeline.
- **Secure internal communication** – Internal communication between different parts of the application should be encrypted, in case that communication is intercepted by an unauthorized person.

Distributed data

The system should be able to handle large volumes of data. What is more, data requirements for each running part should be defined. If data about the same entity is kept in different parts of the system, a method of ensuring the consistency of the data should be developed. What is more, during the development of the system, GDPR has to be taken into account by identifying the sensitive data and methods of keeping it confidential.