## Functional Requirements Document (FRD)

### Project Title: Monthly Bread Pre-Purchase and Daily Collection System

### Prepared By: Thembile and Nkosinathi

### Date: 24th June 2024

### 1. Introduction

1.1 Purpose

The purpose of this document is to outline the functional requirements for the development of a software system that allows residents to pre-purchase bread at the end of the month and collect it daily from local spaza shops. The system aims to provide convenience for residents and ensure steady business for spaza shop owners.

1.2 Scope

This document covers the requirements for the user interfaces, core functionalities, and system interactions. It includes detailed descriptions of features for residents and spaza shop owners, payment integration, notification systems, and reporting mechanisms.

1.3 Definitions, Acronyms, and Abbreviations

* Resident: A user who pre-purchases bread and collects it daily.
* Spaza Shop Owner: A user who sells bread and manages daily collections.
* FRD: Functional Requirements Document.

### 2. System Overview

The system will consist of a web application and a mobile application for residents and spaza shop owners. The primary functionalities include account management, pre-purchase of bread, daily collection tracking, notifications, reporting, and payment processing.

### 3. Functional Requirements

3.1 User Account Management

3.1.1 Resident Registration and Login

* The system shall allow residents to register using their email, phone number, and a password.
* The system shall provide a login interface for registered residents.

3.1.2 Spaza Shop Owner Registration and Login

* The system shall allow spaza shop owners to register with their shop details.
* The system shall provide a login interface for registered spaza shop owners.

3.1.3 Profile Management

* Residents and spaza shop owners shall be able to update their profile information.

3.2 Pre-Purchase System

3.2.1 Bread Pre-Purchase

* The system shall allow residents to select the number of breads to pre-purchase at month-end.
* The system shall display the total cost for the selected number of breads.
* The system shall integrate with a payment gateway for processing payments.

3.2.2 Payment Confirmation

* The system shall confirm payment and update the resident's account with the pre-purchased bread balance.

3.3 Daily Collection Tracking

3.3.1 Spaza Shop QR Code

* Each spaza shop shall have a unique QR code that residents can scan.

3.3.2 Scanning and Selection

* Residents shall scan the spaza shop’s QR code using their mobile app.
* The system shall present the resident with options to select the type of bread (e.g., white, brown, whole grain) and whether it is a full loaf or half loaf.

3.3.3 Purchase Request and Seller Approval

* The system shall send a purchase request to the spaza shop owner’s device.
* The spaza shop owner shall receive a pop-up notification with details of the resident's selection.
* The spaza shop owner shall approve or reject the purchase request.

3.3.4 Purchase Confirmation

* Upon approval by the spaza shop owner, the system shall confirm the purchase and generate a digital proof of purchase for the resident.
* The system shall notify the resident of the successful purchase.

3.3.5 Verification

* The resident shall show the digital proof of purchase to the spaza shop owner.
* The spaza shop owner shall verify the proof of purchase and hand over the selected bread.

3.3.6 Balance Update

* The system shall update the resident’s account balance accordingly after the purchase.

3.5 Notifications

3.5.1 Balance Notifications

* The system shall send notifications to residents when their bread balance is low.

3.5.2 Collection Reminders

* The system shall send daily reminders to residents to collect their bread.

3.5.3 Purchase Notifications

* The system shall notify the spaza shop owner of a purchase request and subsequent approval or rejection.
* The system shall notify the resident of the successful purchase after approval.

3.6 Reporting

3.6.1 Resident Reports

* The system shall provide residents with a history of their bread collections and remaining balance.

3.6.2 Spaza Shop Owner Reports

* The system shall provide spaza shop owners with reports on bread distribution and resident usage.

3.7 Payment Processing and Settlement

3.7.1 Payment Frequency Options

* The system shall allow spaza shop owners to choose their preferred payment frequency (weekly, bi-weekly, or monthly).

3.7.2 Transaction Tracking

* The system shall maintain a detailed record of all transactions for auditing and reconciliation purposes.

3.7.3 Settlement Processing

* The system shall track all transactions made by residents at each spaza shop.
* The system shall calculate the total amount due to each spaza shop owner based on the selected payment frequency.
* The system shall generate a payment summary and transfer the due amount to the spaza shop owner's registered bank account or preferred payment method according to their selected schedule.

3.7.4 Multiple Spaza Purchases

* The system shall allow residents to purchase bread from multiple spaza shops.
* The system shall manage the balance reduction and transaction recording for each spaza shop separately.

### 4. Non-Functional Requirements

4.1 Performance

* The system shall support simultaneous usage by multiple residents and spaza shop owners without performance degradation.

4.2 Security

* The system shall ensure secure authentication and authorization mechanisms.
* The system shall protect user data using encryption and secure storage practices.

4.3 Usability

* The system shall provide a user-friendly interface for both web and mobile applications.

4.4 Scalability

* The system shall be scalable to support an increasing number of users and transactions.

### 5. System Interfaces

5.1 User Interfaces

* The system shall provide a web interface for desktop users.
* The system shall provide a mobile interface for Android and iOS users.

5.2 External Interfaces

* The system shall integrate with external payment gateways such as PayPal or Stripe.
* The system shall integrate with banking APIs for settlement processing.

### 6. Assumptions and Constraints

6.1 Assumptions

* Users have access to internet-enabled devices.
* Spaza shop owners have the capability to scan QR codes or enter collection codes.

6.2 Constraints

* The system must comply with local data protection regulations.
* Payment processing fees should be minimized to reduce costs for users.