

## **Додаток А**

### **Тест план - виробництво морозива**

#### **1. Вступ**

##### **1.1 Мета**

Метою цього документа є детальна розробка заходів з контролю якості, необхідних для виконання проекту “Виробництво морозива”; визначити стратегію тестування та підхід до тестування; визначити сферу діяльності та визначити відповідальність

##### **1.2 Загальний огляд**

Додаток призначений для контролю та менеджменту роботи виробництва з виготовлення морозива. Програма надає інтерфейс для управління персоналом, складськими елементами та елементами виробництва.

#### **2. Тестові елементи**

Необхідно перевірити наступні основні компоненти системи:

- Аплікація;
- Продуктова документація

#### **3. Тест стратегії**

Тестування проводиться з метою перевірки та узгодженості між вимогою клієнтів та готовим продуктом і підготовки інформації про рівень узгодженості. Ця інформація буде надана за допомогою проведення тестування, що проводиться для виявлення дефектів зареєстрованих в результаті цього тестування. Всі дефекти будуть мати пріоритетний атрибут, який визначається шляхом аналізу важливості та частоти використання функціональних можливостей замовником та ризику його появи в операційному використанні.

Дослідницьке тестування буде використане для виявлення залежностей у функціональності та використання цих знань у розробці та тестування тестів.

##### **3.1 Підходи до тестування**

Так як процес розробки має конкретний період, тестування проводитиметься після введення до системи нових функціональних блоків і на основі результатів тестування буде вирішено чи новий блок може бути використаний у парі з готовим продуктом.

##### **3.1.1 Види тестувань**

В рамках проекту будуть виконані наступні типи тестування:

#### **Смоук тестування (Smoke testing)**

Мінімальний набір тестів на явні помилки. Цей тест зазвичай виконується самим програмістом. Програму, що не пройшла такий тест, не має сенсу передавати на глибше тестування.

## **Регресивне тестування**

Виявляє помилки у вже протестованих ділянках початкового коду. Такі помилки — коли після внесення змін до програми перестає працювати те, що мало б працювати, — називають регресивними помилками.

## **Функціональне тестування**

Перевіряє, чи реалізовані функціональні вимоги, тобто можливості ПЗ в певних умовах вирішувати завдання, потрібні користувачам. Функціональні вимоги визначають, що саме робить продукт, які завдання вирішує.

## **Тестування користувацького інтерфейсу**

Виконується QCEs, щоб перевірити, чи відповідає графічний інтерфейс загальним стандартам інтерфейсу користувача та конкретним проектом.

## **Тестування зручності використання**

Виконується з метою визначення зручності використання ПЗ для його подальшого застосування. Це метод оцінки зручності продукту у використанні, заснований на залученні користувачів як тестувальників, випробувачів і підсумовуванні отриманих від них висновків.

### **3.2 Планування тестувань**

#### **3.2.1 Процес тестування**

Під час планування виконуються такі завдання:

- Менеджмент персоналу. Аналіз удосконалень;
- Визначення ознак що підлягають тестуванню;
- Визначення ознак що не підлягають тестуванню

#### **3.2.2 Тест планування критерії стоп продовжити**

Планування тестувань припиняється, коли під час планування тесту виявляються численні розбіжності та суперечливі питання у вимогах.

### **3.3 Тест дизайн**

Тест кейси будуть розроблені лише для UI тестування та функціонального тестування оскільки неможливо точно скласти тест дизайн для smoke тестування та регресивного тестування.

#### **3.3.1 Тест дизайн планування**

Кожен тест повинен задовільняти наступним вимогам:

- Містити загальний опис
- Містити вимоги для чого тест написаний
- Мету тесту

#### **3.3.2 Тест дизайн критерії стоп продовжити**

Тест дизайн припиняється, коли багато розбіжностей і суперечливих питань у вимогах виявляються на будь-якому етапі етапу проектування тесту.

Тест дизайн продовжується коли питання уточнюються з клієнтом і оголошуються команді проекту

### 3.3.3 Тестове середовище

Оскільки основу аплікації складає web інтерфейс – тестування повинно проводитись на різних платформах, у різних браузерях, на різних операційних системах. Для тестування необхідні такі інструменти як: браузер, request handler – для відслідковування надісланих запитів (Google chrome dev tools).

## 4. Ролі та обов'язки

QC Lead - контроль роботи; управління командою; планування тестувань; розподіл роботи.

QC Engineer – аналіз вимог; тест дизайн; тест планування; пошук дефектів