TEST PLAN FOR:

FOOD ORDERING APPLICATION

GROUP: 37

Changelog

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | By | Description |
| 1.0 | 12.12.2021 | Joona Tulisalo | Final version |
| 0.1 | 10.12.2021 | Duong Pham | First version |

# Terms/Acronyms

Terms and acronyms used in the project

|  |  |
| --- | --- |
| Term | Definition |
|  |  |

# Introduction

The Online Food Ordering System described in this paper was created to fill a specific need in the market by allowing small restaurants to offer their clients an online ordering option without having to invest a significant amount of time and money in custom software development. The technology, which is extremely adaptable, enables restaurant personnel to effortlessly manage site material, most notably the menu, via a user-friendly graphical interface. Testing was done using agile methodology to suit our client’s needs. Tests were done to make the best possible software before the deadline.

## Scope

### In Scope

User System:

* Register new user
* Log in to system
* Navigate restaurants’ menu
* Order an item and add to shopping cart
* Remove an/all item from order
* Payment method
* Tracking order status
* View order history

Management System:

* Create new restaurant
* Add new items to menu
* Update items information: price, description, image, etc.

### Out of Scope

All Features will be tested

## Quality Objective

* Find failure and defects
* Verify requirements
* Validate test object
* Evaluate work product

## Roles and Responsibility

* Functional Test: Jani Väisänen
* Database Test: Valtteri Göös
* Usability Test: Joona Tulisalo
* Security Test: Duong Pham

# Test Methodology

## Overview

Our focus in this project was to get a working software. To ensure this, agile programming was being used for testing. As soon as something was not working properly fixes had to be made.

## Test Levels

Integration Test: Top-Down

## Bug Triage

Bugs that are critical to the progress of the software are prioritized first.

## Suspension Criteria and Resumption Requirements

Suspension criteria:

* Making fixes to the software
* Working on a solution

Resumption requirements:

* A possible solution has been made
* Logging information to see what the problem is

## Test Completeness

Tests are completed when a particular feature is working correctly. Bugs are also being handled immediately upon being recognized.

# Test Deliverables

Test cases sheet: In Github

Bug report sheet: In Github

# Resource & Environment Needs

## Test Environment

Google Chrome

Brave browser

## Testing Tools

* Visual studio code
* React.js
* JavaScript
* Heroku
* Node.js
* Windows
* MySQL