HUrricane

Software Test Report

1. **Introduction**

## Purpose and Scope

The purpose of this document is to evaluate how well the developed software meets the specified requirements through testing and to identify potential defects before the final delivery. The Software Test Report covers the testing activities, preparation of the test environment, and evaluation of the results based on verification criteria. In addition, this document presents detailed analyses along with the outcomes of the test scenarios conducted for each requirement, aiming to verify the accuracy and stability of the project.

## Descriptions and Abbreviations

| *STR* | *Software Test Report* |
| --- | --- |
| *UC*  *TC*  *SUT*  *QA*  *ID*  *Req.* | *Use Case*  *Test Case*  *System Under Test*  *Quality Assurance*  *Identifier*  *Requirement* |
|  |  |

# Descriptions of Test Activities

## Document Security

This document is classified and intended exclusively for individuals directly involved in the project. It must not be distributed, copied, or modified by anyone outside the authorized team. Editing rights are strictly reserved for the HUrricane development group to ensure content integrity. The Software Test Report is stored in a secure, access-controlled environment—hosted on encrypted cloud storage and version-controlled through a private repository. To uphold academic integrity and prevent unauthorized disclosure, access credentials are limited and monitored. This document remains valid only during the active phase of the BBM 384 course and will be archived once the course concludes.

## Data Recording, Reduction and Analysis

The data recording, reduction, and analysis procedures to be used during and after the tests defined in this document will be conducted through the **Test Case Definition** document. Each test scenario will be manually recorded through the predefined **Test Case** tool, and each stage of the test will be tracked using this tool. Test outputs will be directly compared with the expected results outlined in the corresponding **Test Case** document. Any deviations, errors, or unexpected behaviors will be manually noted and categorized.

Test results will be stored securely in the project's documentation archive using the **Test Case** tool, ensuring that all data is preserved for future analysis.

This approach ensures a clear connection between each test scenario and the observed outcomes, maintaining consistency and transparency throughout the testing process.

# Test Preparation

## Hardware Preparation

All the testing was done on a desktop computer with a reliable internet connection. The desktop computer has a **Lenovo Ideapad 520** with an **Intel(R) Core(TM) i5-7th Gen CPU @ 2.50GHz** processor and **12.0 GB RAM**.

## Software Preparation

Testing is done with the client running on a **Windows 10 Pro Version 22H2** operating system and browser **Google Chrome Version 112.0.5615.138 Official Build 64-bit**. The system also includes a **Node.js** environment for executing server-side JavaScript code.

A local **PostgreSQL** database server is used for data storage, and the default embedded **Spring Boot web server** is used for handling HTTP requests.

**Instructions for Loading and Initializing Software:**

1. To start the application, first initialize the **PostgreSQL database** and ensure it's running.

Then, to start the backend, run the following command to launch the **Spring Boot server**:  
  
mvn spring-boot:run

1. This command starts the Spring Boot application, which serves the API requests.

Next, start the frontend by running the following command in the **Node.js** environment:  
  
npm start

1. This will initialize the client-side application and ensure the testing environment is ready to receive requests and serve data.

These steps are necessary to prepare the software environment and ensure that both the backend and frontend are ready for the testing procedures.

## Other Pre-test Preparations

Before the testing begins, several preparatory steps will be undertaken:

1. **Database and Server Setup**: The database and server environments will be installed and configured to ensure proper functionality during the testing process.
2. **External Services and Connections**: Any required external services, such as APIs or web services, will be set up and tested for connectivity to ensure seamless integration during testing.
3. **Hardware and Software Requirements Verification**: A thorough check will be performed to verify that all necessary hardware and software requirements are met for the test environment.
4. **Network and System Configuration Check**: The accuracy of the network connections and the system configurations will be validated to ensure proper communication between components during testing.
5. **Review of Test Scenarios and Test Cases**: All test scenarios, test cases, and the overall test plan will be reviewed to ensure completeness and alignment with the objectives of the testing process.
6. **Test Results**

| Use Case Number | | | Use Case Name | Test Case Description  File Name | Retest? | Number of Activity Paths Tested | Number of Non-working Activity Paths | The time spent (in minutes) |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | UC01:Register an Account | | | TC1-A1:Creating an account with a valid email and password  TC1-A2:Creating an account with an invalid email and password | yes | 2 | 0 | 1 |
| 2 | UC02:Change Password | | | TC2:The user changes their password. | yes | 1 | 1 | 1 |
| 3 | UC03:Order Delivery | | | TC3-A1:Order tracking when the courier accepts the order TC3-A2: Order tracking when the courier rejects the order | yes | 2 | 0 | 2 |
| 4 | UC04:Manage Food Item | | | TC04-A1:When the restaurant edits the meal TC04-A2:When the restaurant edits the meal TC04-A3: When the restaurant deletes the meal | yes | 3 | 0 | 3 |
| 5 | UC05:Manage Payment Method. | | | TC05-A1:When the customer adds a payment method  TC05-A2:When the customer edits a payment method  TC05-A3:When the customer deletes a payment method | yes | 3 | 0 | 2 |
| 6 | UC06:Create an Order | | | TC06-A1:When the customer successfully places an order TC06-A2: When the customer's payment method is declined TC06-A3:When the customer cannot find the desired meal or restaurant | yes | 3 | 0 | 3 |
| 7 | UC07:Login | | | TC07-A1:When the user logs in with a valid email and password  TC07-A1:When the user attempts to log in with an invalid email and password | yes | 2 | 0 | 2 |
| 8 | UC08:Customer Order Cancellation | | | TC08-A1When the customer cancels an order with a refund within 3 minutes  TC08-A2When the customer cancels an order with a refund after 3 minutes | yes | 2 | 2 | 3 |
| 9 | UC09:Manage Location | | | TC09-A1:When the user adds a location  TC09-A2:When the user edits a location  TC09-A3:When the user deletes a location | yes | 3 | 3 | 3 |
| 10 | UC10:Logout | | | TC10:When the user logs out of their account | yes | 1 | 0 | 1 |
| 11 | UC11:Update Courier Status | | | TC11-A1:When the courier changes their status  TC11-A2:When the courier's status automatically changes due to order cancellation | yes | 2 | 0 | 2 |
| 12 | UC12:Customer Browses Restaurants and Food Items | | | TC12-A1:When the customer finds what they are looking for  TC12-A2:When the customer makes an invalid search | yes | 2 | 0 | 3 |
| 13 | UC13:Customer Saves Restaurants and Food Items | | | TC13-A1:When the customer saves a meal or restaurant that is already added  TC13-A2:When the customer saves a meal or restaurant that is not already addedTC | yes | 2 | 0 | 3 |
| 14 | AUC1:Approve or Reject Restaurant/Courier Registrations | | | ATC1-A1:When the admin accepts a support request  ATC1-A2:When the admin rejects a support request | yes | 2 | 2 | 2 |
| 15 | UC15:Customer Rates and Reviews Restaurants and Couriers | | | TC15:When the customer views previously saved restaurants | yes | 1 | 0 | 1 |
| 16 | AUC2:Review Management | | | ATC2:When the admin views all past activities of users | yes | 1 | 1 | 2 |
| 17 | AUC3:Manage Customer Support Requests | | | ATC3:When the admin views customers' complaints and requests | yes | 1 | 1 | 3 |
| 18 | UC18:Customer Accesses Order History | | | TC18: When the customer views their past orders | yes | 1 | 0 | 1 |
| 19 | AUC4:Manipulate User Info | | | ATC4-A1:The admin manages a user's information.  ATC4-A2:The admin can deactivate a user's account. | yes | 2 | 1 | 2 |
| 20 | AUC5:View Orders | | | ATC5-A1:When the restaurant views its own orders.  ATC5-A2:When the admin views a restaurant's orders | yes | 2 | 0 | 2 |
| 21 | AUC6:Assign or Reassign Couriers | | | ATC6-A1:When the admin assigns a courier to an order.  ATC6-A2:When the admin assigns the on-duty courier to another order. | yes | 2 | 0 | 4 |
| 22 | UC22:Customer Requests Refunds or Reports Orders | | | TC22:The registered customer requests a refund or reports an issue with an order. | yes | 1 | 0 | 3 |
| 23 | UC23:Courier Registers with Restaurants | | | TC23-A1: The courier registers with restaurants  TC23-A2: The courier attempts to register again while already being registered. | yes | 2 | 2 | 2 |
| 24 | AUC7:Modify Order Status | | | ATC7:When the admin changes the status of an order. | yes | 1 | 1 | 2 |
| 25 | AUC8:View Order and Order History (Search & Filter) | | | ATC8:The admin views and searches orders and order history based on various filters such as date, status, customer, restaurant, and courier. | yes | 1 | 0 | 1 |
| 26 | AUC9:View Users (Search, Filter, Track, Monitor) | | | ATC9:The admin searches, filters, tracks, and monitors user activity based on role, status, registration date, and other criteria. | yes | 1 | 1 | 2 |
|  |  | TOTAL | | ***26*** |  | ***46*** | ***14*** | ***56*** |

## Criteria for Evaluating Results

**a.** The information transmission time between any two objects in the system (e.g., Courier to Restaurant, Restaurant to Customer) should not exceed 30 seconds. Any time above 30 seconds will be considered a failure.

**b.** The minimum number of combinations or alternatives of input and output conditions that constitute an acceptable test result should be defined. For example, an order should be successful when three conditions are met: a valid customer, a valid meal, and a valid payment method. If any of these conditions are not met, the order will be considered failed.

**c.** The maximum or minimum allowable test duration in terms of time or number of events should be defined. For instance, the time to create an order should not exceed 2 minutes. If it takes longer than 2 minutes, a performance issue is assumed. Similarly, canceling an order should not take longer than 2 minutes.

**d.** The maximum number of interrupts, halts, or other system breaks that may occur should be specified. For instance, only one interruption is acceptable during the test. If two or more interruptions occur, the test will be considered a failure.

**e.** The allowable severity of processing errors should be determined. For example, a payment transaction error should not cause the system to crash. However, if the payment amount is displayed incorrectly, the transaction will be marked as failed, and the error will be logged.

**f.** Conditions under which the result is inconclusive and re-testing is to be performed should be clarified. For instance, if the delivery status of an order is not updated correctly, or if the system shows an incorrect error message, this will be considered inconclusive, and the test should be re-executed.

**g.** Conditions under which the outputs are to be interpreted as indicating irregularities in input test data, in the test database, data files, or in test procedures should be defined. For example, if the customer data used during the test is incomplete or incorrect, and this causes the system to fail, the test will be invalid, and it should be re-executed with corrected data.

**h.** The allowable indications of the control, status, and reports of the test, and the readiness for the next test case should be provided (may be the output of auxiliary test software). For instance, the test software should indicate whether each test has been completed and provide an approval for the next test to begin. If the test reports are incomplete or errors are present, the next test should not be started.

**i.** Security tests should be conducted during the testing process. If any security vulnerability is found, the test will be considered failed, and appropriate security measures should be taken. Also, user experience (UX) tests should be performed, and if the user interface is difficult to use or faulty, it should indicate a failure.

# References

* HUrricane Software Requirement Specifications
* HUrricane Test Case Definitions
* HUrricane Use Case Definitions