Date: 01/01/2024

**Online Food Ordering Portal**

1. **Objective:**

* The main objective is to provide a friendly environment to order any edible item easily and efficiently.
* There should be no difficulties in preparing different reports.
* The new users or existing users should not be able to face any problems regarding any order.
* Every user should be able to place order or cancel it.
* The Restaurant manager should have access to get all details about all previous order and all orders currently processing in the restaurant.
* The owner should have access to each and every user’s account.
* It should ensure data security and privacy of all users.
* It should enhance the user experience by making a user-friendly interface.
* Everyone should be able to make payments easily.

1. **Scope:**
2. **Within the Scope:**

* Login Management: There should be a proper login system. It should be validated that both Manager and users are able to login into the system with their own respective ID and passwords. There should be no problem in that.
* Search Management: There should a specific search platform in which users are able to search for any particular item by just filling its name into the filter tab.
* User Management: In this type, each user should have the facility to check that which order has been done by the user and when it was done. There should be every information about the user in it.
* Resource Management: In resource management, it includes the type of items present there. It should be mainly accessible to the manager to keep a track of record of all the items and the orders currently going on in the system.
* Account Management: The Manager should have access to all the user’s account and it should have the power to remove any user account from the system. There should also be a report option for the users so that they can report any incident.
* Bill Management: There should be an easy and friendly way for users to make their payments easily and keep a track of it.

1. **Out of the scope:**

* Non-functional Testing like stress, performance is beyond the scope in this project.

1. **Test Strategy:**
2. **Levels of Testing:**
3. Unit Testing: The developer has to validate the code according to the given syntax and semantics.
4. System Testing: Users should be able to perform end to end operations like registration, placing a new order, viewing user’s details, updating new items in the menu, Book a Table, Giving feedback and customer support.
5. Integration Testing: We have to check whether the user is able to use the external web browsers such as Microsoft Edge, Google Chrome or Mozilla Firefox.
6. User Acceptance Testing: This will allow the user to perform end to end real world scenario’s and the customer’s expectations should be fulfilled.
7. **Types of Testing:**
8. White Box Testing: The validation of code is done.
9. Black Box Testing: There should be features like ordering, booking tables, adding & cancelling dishes, giving feedback & making payments. These should be ensured from the user’s point of view.
10. Smoke Testing: The development lead needs to validate the application and check whether it is ready to undergo detailed testing or not.
11. Regression Testing: It should validate that the new changes should not affect the existing functionalities in a negative way.
12. Retest: Once the defect is closed, the test engineer will should perform testing again in order to confirm that the problem is fixed.
13. **Test Design Techniques:**
14. **White Box Testing techniques:**

* Statement Testing: It should validate all the statements at least once in order to avoid any syntax or semantic error.
* Branch Testing: Validate all the branches at least once.

1. **Black Box Testing techniques:**

* Decision Table: At the time of placing order, it should show each and every detail regarding that item at the same time.
* ECP & BVA: This should be used while user is performing registration actions.
* State Transition Testing: It should allow the user to perform different end to end operations like ordering, viewing food items, etc.

1. **Terminologies:**

* **Menu:** The process of maintaining a record of all resources including food items and other materials.
* **Address:** It asks the exact address of the user at which the order will be delivered.
* **Order:** It is used to order any item you need. You can directly order or it can be added to the cart.
* **Cancel:** This will be used when a user wants to cancel any order which is present in the user’s cart before placing order.
* **Payment:** There will only Cash on delivery on every order.
* **Database:** We have three databases for book a table, billing details & feedback form.

1. **Configuration Management Tool:**  This tool allows you to store all the project related documents at a single place like Git-Hub or Git-Lab.
2. **Area plan for automation:** It should list all the test cases/ functionalities/ modules planned for automation testing.
3. **List of Automation tools:** Selenium is the only tool we are using for automation testing.
4. **Entry and Exit criterion:**

**Entry criteria:**

* The requirement should be approved.
* Test cases should be reviewed.
* Test environment should be compatible.
* Test data should be ready.
* Entry criteria is done after the code implementation is done.

**Exit criteria:**

* The passing rate should be above 95%.
* There should be no critical defects.
* The code should not have any syntax or semantics error.
* Integration should work properly.
* The system should be allowed to perform different end to end operations.

1. **Test Environment:**

* Windows 11
* 16 GB RAM
* 512 GB SSD
* Processor AMD Ryzen 9
* Chrome, Edge, Firefox.

1. **Test Deliverables:**

* Test Plan: A document for outlining the testing approach, strategies, scope, objectives and schedules for Restaurant Management.
* Test Cases: Each testing scenario, including positive and negative test cases, to be executed during testing.
* Test Data: Data sets are used for testing, including booking table, placing orders and data related number of dishes present.
* Test Execution Reports: The detailed execution of test cases, including test results, identified defects and any deviation from expected behavior.
* Defect Reports: The detailed reports documenting of identifies defects, including their severity, steps to reproduce and potential impact on the system.
* Report on Test Summary: A brief summary report providing an overview of testing activities, results and any issues. It includes into overall quality of the Restaurant’s Portal.

1. **Hiring and Training:**

* Hire a manual tester with 2+ year experience and have data management domain knowledge.
* Hire a security tester with 6+ years of experience.
* Providing the training on the specific domain.

1. **Risks and Mitigations:**
2. **Risks:**

* There can be slow server response times.
* Fault in process of booking tables and placing orders.
* There can be system glitches.
* Issuance with different browsers.

1. **Contingency:**

* Users can easily place orders from restaurant.
* Users can easily clear the cart.

1. **Mitigations:**

* Keep the Restaurant portal up-to-date with latest security patches.
* Establish reliable data backup and recovery procedures to minimize the impact of data loss or corruption.

1. **Schedule:**

|  |  |
| --- | --- |
| **Activity** | **Duration** |
| Requirement Analysis | 25 man hours |
| Test case preparation | 45 man hours |
| Test case review | 20 man hours |
| Test data preparation | 20 man hours |
| Test execution | 60 man hours |
| Retest | 20 man hours |
| Regression Test | 40 man hours |
| Test Closure | 20 man hours |

1. **Roles & Responsibilities:**

|  |  |
| --- | --- |
| **Roles** | **Responsibilities** |
| Admin | * Checking placed orders. * Order cancellation and details. |
| User | * Already placed orders can be viewed by the user. * User can book a table. * User can review the details of food. * User can make payments. |
| Test Lead | * Preparing the Test Plan. * Reviewing the test cases. * Reviewing the test data. * Reviewing the test environment. * Aligning the task to the team. |
| Test Engineer | * Reviewing requirements. * Reviewing test plan. * Preparing test cases. * Preparing test scenarios. * Preparing test data. * Reviewing test environment. |

1. **Approval information:**

|  |  |
| --- | --- |
| **Document** | **Approving Rule** |
| Requirements | SME/Stakeholder |
| Test Plan | Project Manager |
| Test Cases | Test Lead |
| Defects | Developer |
| Automation Script | Automation Lead |

1. **Assumptions:**

* Users are assumed to have stable internet connections for accessing the Restaurant’s online food delivery portal.
* The portal assumes compatibility with common devices and browsers and users are expected to use up-to-date and compatible software.
* Users are expected to adhere to restaurant’s rules and guidelines, including all important informations.

1. **Test Metrics:**

* Passed test cases percentage: (no. of passed test cases/no. of test cases executed) \*100
* Failed test cases percentage: (no. of failed test cases/no. of test cases executed) \*100
* Fixed defect percentage: (defects fixed/defects reported) \*100
* Accepted defect percentage: (defects accepted as valid by dev team/total defects reported) \*100
* Defect deferred percentage: (defects differed for future releases/total defects reported) \*100
* critical defects percentage: (critical defect /total defects) \*100