**PARKING CALCULATOR – TEST PLAN**

**1. Introduction**

The goal of this document is to provide a detailed overview of the testing activities inside Parking Calculator project, including test strategy, test execution, results, and any defects identified. This document covers functional testing, cross-browser testing, and edge case testing for Parking Calculator project.

**2. Test Overview**

* Project Name: Parking Calculator
* Tested By: Kevin Sugeng
* Tools:
  + Jira: For tracking project progress, managing test cases, and logging defects.
  + Zephyr Scale: Tools that can integrated with Jira for creating and executing test cases with Jira.
* Test Environment:
  + Web browsers: Chrome, Firefox.
* Epic and User Stories Covered:
  + Epic: Testing Parking Calculator

|  |  |  |
| --- | --- | --- |
| **User Story ID** | **User Story Title** | **Description** |
| SCRUM-10 | As a user, I want to select a parking lot type so that I can calculate the parking cost based on my selection | Ensure users can select different parking lot types and calculated correct calculations. |
| SCRUM-17 | As a user, I want to calculate the cost of Valet Parking based on different durations so I can see how much I need to pay. | Test Valet Parking cost calculations for various durations. |
| SCRUM-18 | As a user, I want to calculate the cost of Lot Parking based on different durations so I can know the total fee. | Verifies accurate Lot Parking fee for different time periods. |
| SCRUM-19 | As a user, I want to calculate the cost of Hourly Parking based on different durations so I can find out what the charge will be. | Checks Hourly Parking charges, making sure correct cost increases with time. |
| SCRUM-20 | As a user, I want to calculate the cost of Hourly Parking based on different durations so I can find out what the charge will be. | Confirms Garage Parking fees are calculate correctly for hourly and daily rate. |
| SCRUM-21 | As a user, I want to calculate the cost of Surface Parking based on different durations so I know how much it will cost me. | Verifies Surface Parking cost calculations are accurate for several durations. |

**3. Test Execution and Cycles**

**Test Cycles Overview**

Test Cycles 1: Functional Testing Cycle 1

* Cycle ID: SCRUM-R6
* Duration: 07/28/24
* Objective: Validate basic functionality of Parking Calculator across different parking lot types
* Test Cases: 5
* Pass: 4
* Fail: 1 (SCRUM-T6 – Valet Parking)
* Summary: In the first cycle, most of test cases has passed except for the valet parking, where the system showed an incorrect parking cost. This issue was resolved by developer and in subsequent testing.



Test Cycles 2: Functional Testing Cycle 2

* Cycle ID: SCRUM-R7
* Duration: 07/29/24
* Objective: Resolved Issues from the first cycle and ensure edge cases are handled correctly.
* Test Cases: 25
* Pass: 16
* Fail: 9

Summary: The second cycle focused on retesting the failed cases from the first cycle and edge cases. . This issue was resolved by developer and in subsequent testing.

**4. Defect Testing Summary** 

**5. Defect Testing Summary**

* Test Cycles Overview:
  + Functional Testing Cycle 1: Focused on validating core functionality like parking lot selection and cost calculation for different parking types. A few issues were identified and fixed by developer and retested.
  + Functional Testing Cycle 2: Aimed at testing specific scenarios like edge cases, date, and time validations, and making sure all previous identified were resolved. This cycles more involved on testing multiple-days iterations and scenarios where exit and entry time is the same.
* Defect Identified:
  + A total of 9 defects were identified during Functional Testing cycle 2 which all related to incorrect parking cost calculations for specific durations and the system did not handle the edge cases properly.
* Defect Resolution:
  + All defect were fixed and re-tested in the same cycle and after retesting, all the test cases passed successfully, confirming that the defects were resolved.
* Conclusion:
  + Parking Calculator application has successfully passed all test cases after resolving the identified defects. The system is functioning as expected, with accurate calculations for all test scenarios, including edge cases.