System/Integration Test Plan

For Spark App

To Get Data From Rest API

|  |  |
| --- | --- |
| Version | 0.1 |
| Release State | Draft |
| Prepared By | Jogananda Borah |
|  |  |
|  |  |

**Document Change Control**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Authors** | **Summary Of Changes** |
| 0.1 | 30/05/2022 | Jogananda Borah | Initial Draft |

* 1. Introduction
  + 1.1 Project Overview
  + 1.2 Purpose of Document
  + 1.3 Document Audience
* 2. System Testing Scope
  + 2.1 In Scope
  + 2.2 Out Scope
* 3. Test Coverage
* 4. Test Environment, Test Data, Test Tool and Test Schedule
  + 4.1 Test Environment & Tool
  + 4.2 Test Data
  + 4.3 Test Schedule
* 5. Process and Management
  + 5.1 Test Deliverables
  + 5.2 Defect Management
* 6. Assumptions
* 7. Risks and Dependencies
  + 7.1 Risks
  + 7.2 Dependencies
* 8. Governance
  + 8.1 Reporting
  + 8.2 Entry Criteria
  + 8.3 Exit Criteria
* 9. Roles & Responsibilities

1. **Introduction**
   1. **Project Overview**

To create as Spark based Application to read data from REST endpoint, get insights and load the top 10 most pedestrians locations by day and month into a .csv file

* 1. **Purpose of document**

The purpose of this test plan document is to:

* Provide an overview of the testing processes and deliverables
* Identify the functionality that will be tested
* Define the entry and exit criteria
* Define testing environment and data requirements
* Communicate major dependencies, risks, issues and assumptions
* Provide a detailed schedule and resource plan
* Define management processes and control measures
* Ensure agreement from all relevant stakeholders about testing scope, approach and timeline
* Communicate roles and responsibilities
  1. **Document Audience**

This document is intended for the following audience

* Project Team
* QA
* Developer
* Business Stakeholder

1. **System Testing Scope**
   1. **In Scope**

Below is the high level test scope considered for testing

|  |  |
| --- | --- |
| Business Feature | Description |
| Get Top 10 (most pedestrians) locations by day insights | Get source data from REST endpoint  Filter and process the data to get the Top 10 most pedestrians locations by day  Write the Top 10 locations with details like Name, Year, Month, Date, Daily Count, Longitude and Latitude to a .csv file |
| Get Top 10 (most pedestrians) locations by month insights | Get source data from REST endpoint  Filter and process the data to get the Top 10 most pedestrians locations by month  Write the Top 10 locations with details like Year, Month, Monthly Count, Longitude and Latitude to a .csv file |

* 1. **Out Scope**

Generally, any item that is not exclusively called out in the In Scope section are considered as out of scope for testing.

1. **Test Coverage**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Scenario | Testing Start Date | Testing End Date | Testing Tasks | Responsible |
| Create Test Database and Tables | 27/05/2022 | 30/05/2022 | * Create Test Database using SQLITE3 in python * Create Test Tables in the above created database  1. Table “**prod\_source\_sensor\_cnt**” for source sensor count data from URL (REST endpoint) “https://data.melbourne.vic.gov.au/resource/b2ak-trbp.json” 2. Table “**prod\_source\_sensor\_loc**” for source sensor location data from URL (REST endpoint) “” 3. Table “**prod\_top\_ten\_daily**” for app generated csv for the insight Top 10 most pedestrians locations by Day 4. Table “**prod\_top\_ten\_monthly**” for app generated csv for the insight Top 10 most pedestrians locations by Month | QA |
| Validate Top 10 (most pedestrians) locations by day insights | 27/05/2022 | 30/05/2022 | * Validate the count of records in the app generated csv, it should be 10. * Validate the data in the app generated csv.  1. **Expected Data**: Get the top 10 mostpedestrians locations by Day using TEST Query from the tables **prod\_source\_sensor\_cnt** and **prod\_source\_sensor\_loc tables** 2. **Actual-Data:** Get the top 10 mostpedestrians locations by Day using TEST Query from the table **prod\_top\_ten\_daily** 3. Compare the **expected** and **actual** data in point 1 and 2, there shouldn’t be any difference. | QA |
| Validate Top 10 (most pedestrians) locations by month insights | 27/05/2022 | 30/05/2022 | * Validate the count of records in the app generated csv, it should be 10. * Validate the data in the app generated csv.  1. **Expected Data**: Get the top 10 mostpedestrians locations by Day using TEST Query from the tables **prod\_source\_sensor\_cnt** and **prod\_source\_sensor\_loc tables** 2. **Actual-Data:** Get the top 10 mostpedestrians locations by Day using TEST Query from the table **prod\_top\_ten\_monthly** 3. Compare the **expected** and **actual** data in point 1 and 2, there shouldn’t be any difference. | QA |

|  |  |
| --- | --- |
| Scenario | Test Query |
| Validate Top 10 (most pedestrians) locations by day insights | 1. **Expected Data:** Select Loc\_Name as Loc\_Name, Year as Year, Month as Month, mDate as mDate, Daily\_Cnt as Daily\_Cnt, Longitude as Longitude, Latitude as Latitude from (Select \*, row\_number() over(order by daily\_cnt desc) daily\_rnk from ( Select cnt.sensor\_id, cnt.sensor\_name as loc\_name,cnt.year,cnt.month,cnt.mdate,loc.longitude,loc.latitude,sum(hourly\_counts) as daily\_cnt from prod\_source\_sensor\_cnt cnt join prod\_source\_sensor\_loc loc on cnt.sensor\_id = loc.sensor\_id group by cnt.sensor\_id,cnt.sensor\_name,cnt.year,cnt.month,cnt.mdate,loc.longitude, loc.latitude order by sum(hourly\_counts) desc)) where daily\_rnk <= 10 2. **Actual Data:** Select Loc\_Name, Year, Month, mDate, Daily\_Cnt, Longitude, Latitude from prod\_top\_ten\_daily order by Daily\_Cnt desc limit 10; |
| Validate Top 10 (most pedestrians) locations by month insights | 1. **Expected Data:** Select Year,Month,Monthly\_Cnt,Longitude,Latitude from (Select cnt.sensor\_id,cnt.sensor\_name,cnt.year,cnt.month,loc.longitude, loc.latitude,sum(hourly\_counts) as monthly\_cnt from prod\_source\_sensor\_cnt cnt, prod\_source\_sensor\_loc loc where cnt.sensor\_id = loc.sensor\_id group by cnt.sensor\_id,cnt.sensor\_name,cnt.year,cnt.month,loc.longitude, loc.latitude order by sum(hourly\_counts) desc) limit 10 2. **Actual Data:** Select Year, Month, Monthly\_Cnt, Longitude, Latitude from prod\_top\_ten\_monthly order by Monthly\_Cnt desc limit 10 |

1. **Test Environment, Test Data, Test Tool and Test Schedule**
   1. **Test Environment & Tools**

**Python Virtual** PipEnv will be used for testing purposes

**Test Database**: belongQE.db

**Excel**: For Defects Management

* 1. **Test Data**

**Expected Source Data:**

Sensor Count:

Sensor Location:

**Actual Source Data:**

Top 10 most pedestrians by Day: .csv file generated by the spark app

Top 10 most pedestrians by Month: .csv file generated by the spark app

* 1. **Test Schedule**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.No. | Activity | Start Date | End Date | Comments |
| 1 | Test Plan | 27/05/2022 | 29/05/2022 |  |
| 2 | Test Execution | 28/05/2022 | 30/05/2022 |  |
| 3 | Test Completion Report | 30/05/2022 | 30/05/2022 |  |

1. **Test Deliverables**
   1. **Test Deliverables**

The key deliverables to support the testing efforts are:

|  |  |  |
| --- | --- | --- |
| Deliverables | Owner | Comments |
| Test Plan | QA |  |
| System Test Cases | QA |  |
| System Test Completion Report | QA |  |

* 1. **Defect Management**

**Defect Triage Meetings**

* Regular (Daily) Defect Triage meeting to be conducted during Test Execution phase to ensure defects are prioritized and fixed in the right order and minimize the impact on Project and Test phases.
* QA team will take a lead role in the Defect Management processes and oversee defect triage.
* Issues requiring external parties involvement would need to be communicated on different channels such as emails and/or this meetings

**Defect Severity Classification**

|  |  |
| --- | --- |
| Severity | Description |
| Sev-1 | * The entire application, component, function will not work and no work-around is available. Testing cannot proceed. * Major impact to testing schedule * Must be addressed as a matter of Highest Priority |
| Sev-2 | * A significant portion of the application, component, function will not work but work-around is available * Testing is severely impacted, though can proceed * Must be addressed as a matter of High Priority |
| Sev-3 | * A minor portion of the application, component, function will not work but work-around is available * Testing is impacted, though can proceed * Must be addressed as a matter of Medium Priority |
| Sev-4 | * The function doesn’t perform quite as expected. Business functionality is not compromised, e.g. Cosmetic issue * Must be addressed as a matter of Low Priority |

1. **Assumptions**

|  |  |  |
| --- | --- | --- |
| # | Assumptions | Mitigation if not met |
| 1 | Source of truth for data is Rest endpoint, no data cleaning required | Alternate plans to cleanse the data as well |
| 2 | Signed off requirements, design, mapping documents | Keep following up with project team, product owner, BA’s and Dev team if required |
| 3 | Resources available at scheduled times | Alternate plans to work additional hours or weekend to catch up any lost time |

1. **Risk and Dependencies**
   1. **Risk**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Risk | Impact(H,M,L) | Mitigation |
| 1 | External REST endpoint is used for reading sensor data | High – No data will be sourced, and application will throw error | Alternate plans to make sure endpoint is always available or fallback to previous state data |
| 2 | External open-source libraries used in the test tool which may be deprecated or not maintained further | High – Tests will not run | Alternate plans to use different libraries or write own code with the assistance of Dev team |

* 1. **Dependencies**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Dependency | Impact(H,M,L) | Mitigation |
| 1 | Code Deployment is not complete 100% | Medium – Testing timelines would be impacted | Following up with Dev Team |

1. **Governance**
   1. **Reporting**

An System/Integration Test summary report will be rolled out to all stakeholders twice in a week during testing phase, additionally a daily test status report will be sent to the BA’s, Product Owner, Dev Team and QA’s involved

* 1. **Entry Criteria**

|  |  |
| --- | --- |
| Test Phases/Test Types | Entry Criteria |
| System/Integration Testing | * Documentation signed off or agreed upon by all stakeholders * Delivery of build items as per plan * 100% Unit tests completed * 0 Unit Test defects * Automated TVT to be complete prior to code deployment into System test environment |

* 1. **Exit Criteria**

|  |  |
| --- | --- |
| Test Phases/Test Types | Entry Criteria |
| System/Integration Testing | * 100% System tests executed * No outstanding Severity 1 or 2 defects * System/Integration Test Completion Report completed |

|  |  |
| --- | --- |
| Role/Name | Responsibilities |
| Project Manager/ Product Owner/Scrum Master | * Engage testing team for test estimates * Manage project timelines, risks and milestones * Review test plans * Review Test Completion Reports |
| BA/Data Analyst | * Review test deliverables and test coverage. * Write Acceptance criteria and DOD * Support Defect/Issue resolution and/or escalation |
| Developer/Data Engineer | * Issue/Defects resolution/escalation/retesting * Review/Execute test coverage * Scope review walkthrough |
| QA | * Provide test efforts and estimates * Assist in writing Acceptance Criteria and DOD * Develop Test Plan for planned test levels * Write, review test scripts and ensure traceability to business requirements and technical documents * Manage defects, raise and retest * Contribute to Project Reporting * Test Execution and check if DOD is met * Test Data Management * Provide Test execution status information, daily status report * Raise risks if any |

1. **Roles & Responsibilities**