Snack Truck Application

Test Plan

Version: 1.0

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 9/21/2019 | 1.0 | Initial Draft | Avni Sood |
| - | - | Reviewed by | - |

**Approval**

|  |  |  |
| --- | --- | --- |
| **Role** | **Name** | **Approval** |
| **Staff Software Dvlpmt Engineer** | **-** | **Approval required** |
| **Project Manager ITS** | **-** |
| **Sr Test Manager** | **-** |
| **Mgr IT Applications** | **-** |

Table of Contents

[1. Introduction 4](#_Toc19798805)

[1.1 Purpose 4](#_Toc19798806)

[1.2 Project Overview 4](#_Toc19798807)

[1.3 Project Features Implemented 5](#_Toc19798808)

[1.4 Example screen 6](#_Toc19798809)

[1.5 Test Objectives 6](#_Toc19798810)

[1.6 Related Documentation 8](#_Toc19798811)

[1.7 Detailed Schedules and Milestones 8](#_Toc19798812)

[2. Test Requirements 9](#_Toc19798813)

[2.1 Tests 9](#_Toc19798814)

[2.2 Test Environment Smoke Test Requirements 9](#_Toc19798815)

[2.2.1 Environment readiness 9](#_Toc19798816)

[2.2.2 Smoke Test 9](#_Toc19798817)

[2.3 Functional Test Requirements 9](#_Toc19798818)

[2.4 Regression Tests Requirements 9](#_Toc19798819)

[2.5 Performance Test Requirements 9](#_Toc19798820)

[2.6 Security Test Requirements 9](#_Toc19798821)

[2.7 Defect Test Requirements 10](#_Toc19798822)

[2.8 UAT Tests 10](#_Toc19798823)

[2.9 Traceability Matrix 10](#_Toc19798824)

[2.10 Testing NOT APPLICABLE: 10](#_Toc19798825)

[2.11 Test Requirements 11](#_Toc19798826)

[2.11.1 Test Types 11](#_Toc19798827)

[3. Resource Requirements 12](#_Toc19798828)

[3.1 Hardware Requirements 12](#_Toc19798829)

[3.2 Software Requirements 12](#_Toc19798830)

[4. List of Outstanding Issues 13](#_Toc19798831)

[5. Defect Management 13](#_Toc19798832)

[6. Test Deliverables 14](#_Toc19798833)

[7. Assumptions/Dependencies/Risk 15](#_Toc19798834)

[7.1 Assumptions 15](#_Toc19798835)

[7.2 Dependencies 16](#_Toc19798836)

[7.3 Risk 16](#_Toc19798837)

[8. Entrance/Exit Criteria 17](#_Toc19798838)

[8.1 Entrance Criteria 17](#_Toc19798839)

[8.2 Exit Criteria 17](#_Toc19798840)

[9. References 18](#_Toc19798841)

[10. Glossary 19](#_Toc19798842)

# 1. Introduction

## Purpose

The purpose of this document is to detail the approach to be followed for the testing of Snack Truck Android application.

The document seeks to outline the scope, schedules, responsibilities, resources, metrics, issues, risks, environment, and data required to accomplish this.

The sign off of this document indicates that the signing reviewers agree with the adopted and stated testing approach in relation to the Snack Truck Application project. The purpose of the Test Approach is to outline the test effort that must be completed to contribute to project sign-off and go-live.

## Project Overview

Snack Truck, which runs food trucks that sell a selection of snacks. Instead of having one line. Snack Truck have a pool of phones and tablets each truck can pass around, so multiple people can order snacks at once.

## Project Features Implemented

Snack Truck Application cover the below list of features and based on the priority, its stated below

**Priority One Use Case:**

* A user can select a set of snacks.
* Pressing a Submit button at the bottom of the UI will finish the order and show the summary (a list of selected snacks) in a dialog.
* Once the order summary dialog is dismissed, all selections will return to default, preparing the app for the next order.

Note: The network service for actually placing the order isn't built yet, so there is a stub & comment where call is put to send the order.

**Priority Two Use Case**

The customer should be able to filter the snack list by snack type.

* There will be two checkboxes at the top of the app, Veggies and Non-Veggies. Both are checked by default.
* When the Veggies checkbox is checked, Veggie snacks are shown in the main list. When the Veggies checkbox is
* unchecked, Veggie snacks aren't shown. Same for Non-Veggie.
* All requirements of the priority 1 use case are met.

**Priority Three Use Case**

The truck operator should be able to add more snacks to the list.

* There must be an “add” action in the action bar at the top of the app
* When a user selects the “add” action, a dialog is displayed.
* The dialog should have a toggle for veggie/non-veggie, and a text field to enter the new snack name
* The dialog should have “Save” and “Cancel” buttons.
* When the user hits “Save”, the dialog is dismissed, and the user’s new snack is added to the list.
* When the user hits “Cancel”, the dialog is dismissed.

Note: Using the app both for menu editing and for customer ordering presents a problem. The spec doesn't call for any mechanism to distinguish an operator, who may both order things and edit the menu, from a customer, who may only order things.

## Example screen



## Test Objectives

Below is the list of various Types of Testing that will be performed in this Project.

QE team will collaborate with the project team to determine the functionality and user stories that will be implemented for the overall release.  QE team will review the user stories.  After review, the QE team will state which user stories will be testable and non-testable and also which can be automated and which cannot. And test/automate the testable user stories

**Unit Testing:** This will be performed by development team in collaboration with QE team and the objective is to ensure each individual unit/component of the application is working as required .And comprises of below tasks:

* + Understand User stories
  + Identify tools for Unit testing
  + Write unit test case
  + Execute tests
  + Setup Continuous Integration

**Functional Testing:** This will be performed by the Functional testing team and the objective is to ensure that the required functional changes are implemented as per the business needs across Snack Truck Application. And comprises below tasks

* + Understand functional changes of application
  + Document, Design & Review the test cases
  + Automate functional tests that are identified as to be automated
  + Test the functional changes and validate the defects
  + Maintain test artifacts
  + Setup CI/CD for automated tests
  + Test Reporting for both Manual and automated tests

**Regression and Integration Testing** – This will be performed by the Functional Testing Team in collaboration with other impacted testing teams and the objective is to test all the integrated system functions correctly work as per the requirements. And comprises below tasks

* + Understand integrated apps
  + Document and Design necessary regression test cases
  + Test the new functional changes does not impact the existing system
  + Automate regression tests that are identified as to be automated
  + Setup CI/CD for automated tests
  + Validate the defects and Maintain test artifacts
  + provide the automation status reports

**Performance Testing:** This will be performed by QE team, their objective is to understand the performance impact of the system and evaluate the application and its failures. And comprises below tasks

* + Understand the requirements
  + Script & design performance tests
  + Test & provide the Performance status reports
  + Raise and validate the defects

**User Acceptance Testing (UAT) –** User Acceptance testing to be performed by Business users of Snack Truck Application after the QA Sign Off and need to certify, whether all their business needs are met or not.

## Related Documentation

XYZ is the tool used by functional testing teams for test management. All the test cases are created within Snack Truck repository and linked to appropriate user stories and tags.

## Detailed Schedules and Milestones

***Schedule & Timelines***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Feature** | **Sub Feature** | **Target Dates** | |  |
| **Development** | **QA Date** | **PROD Deploy** |
| *Submit order* | *The customer should be able to select snacks from the list, and submit orders.* | *8/19* | *9/19* | ***10/12*** |
| *Filter snack* | *The customer should be able to filter the snack list by snack type.* | *8/30* | *9/30* | ***11/10*** |
| *Add Snack* | *The truck operator should be able to add more snacks to the list* | *9/15* | *10/15* | ***12/28*** |

# Test Requirements

## Tests

The testing will be performed for different requirements (mentioned in the section 1.3) which addresses different functionalities involving snack selection and snack addition for both veggie and non-veggie items.

## Test Environment Smoke Test Requirements

In order to start testing activities in the Test environment, the following requirements should pass, failing which would delay starting of test execution or even non acceptance of the build.

## Environment readiness

The purpose of this testing is to make sure the application is stable and properly integrated and launched.

## Smoke Test

This will be performed by QE Team. After a new build, the QE Team will run a Smoke/ Sanity Test suite to make sure basic functionality is working.

This will be covering the high-level functionalities, Launch application, verify snack list is shown, verify new snack is added successfully, verify snack selection is done.

## Functional Test Requirements

QE team has developed test cases to test the functionality of all the new features implemented along with regression of existing features. These test cases will be executed in Test environment for each build that is being delivered to testing team.

These test cases should be able to validate the integration of different interfaces with each other and will ensure data flow from each application is uninterrupted.

## Regression Tests Requirements

QE team has identified regression test cases for the project which should be executed every time we receive a new build for the project. All the automated test case will be executed and identified regression will be executed. This covers the critical and defect prone areas in the application identified in the iteration related to the features being tested.

Team will run the regression tests mostly in conjunction with the functional test cases and the order of the execution will be prioritized based on the situation and release time of different builds.

## Interrupt Test requirements

QE team has identified test cases for interrupt scenarios. Those interruptions can be; incoming and outgoing SMS/MMS/calls, incoming notifications, battery/cable insertion and removal for better uses, network outage and recovery, switch off/switch on of the media player and other connecting devices, Low memory warning, and device power cycle (like; low battery notification).

## Installation Testing

QE team has identified test cases for testing if application is installing, uninstalling and updating properly without interruptions. All cases will run on all mentioned devices and versions.

## Performance Test Requirements

QE team has identified and developed test cases for the project which should be executed. All the scripts are ready and executed based on Load and stress requirements on the provided environment. All scenarios must be covered and identified defects are logged.

## Security Test Requirements

QE team has identified all security threats on the application and executed tests based on the requirements. All scenarios must be covered and identified defects are logged

## Defect Test Requirements

Any defects or change requests identified during the course of testing will go through defect prioritization and the Business stakeholders will decide which defects need to be fixed during this release. QE team will need to contribute to validate these fixes.

## UAT Tests

User Acceptance tests will be run by Business SMEs. They will prepare the Acceptance Use Case scenarios for different functionality end to end flow. The purpose of this tests is to validate that the system is acceptable based on the requirements given by Business during the beginning of the project and can be deployed to production.

## Traceability Matrix

This matrix contains high level overview for all the requirements for Snack Truck test cases coverage details corresponding to each acceptance criteria.

All the Functional test cases for this Project are captured based on the acceptance criteria.

## Testing NOT APPLICABLE:

**Network service testing:** Application does not contain any service to actually place the order so Service testing is not in scope.

**Operator/User based testing:** Application is not developed based on the user/operator. So no testing around operator use will be performed.

## Test Requirements

Snack Truck Application’s test requirements will be created as user stories and maintained within Test management tool .

*Functional Testing field with in each User story defines the scope of the user story for functional testing team and it is defined with the below status values*

|  |  |
| --- | --- |
| **Functional Testing**  **Status** | **Definition** |
| Blocked | User story Blocked due to various reasons and could not proceed further |
| Completed | Selected when Functional testing is complete for the user story |
| In Progress | Selected Functional team is working on it actively |
| Non Testable | User story that are not related to QA |
| Not Started | Functional QA team is yet to START working on User story |
| Not Applicable | User story Not Applicable to Functional QA team |
| Not in Scope | User story that got created during requirement gathering phase, but later advised that it will **NOT** be part of release |

## Test Types

The following test will be performed during the test execution phase

* Unit testing [By Dev team]
* Smoke Testing [by QE team Testing team]
* Functional Testing [by QE Testing team]
* Regression Testing [by QE Testing team]
* User Acceptance Testing [by QE Testing team]
* Business Flow testing [End to End Testing team]
* Performance & Load testing [by QE testing team]

# Resource Requirements

## Hardware Requirements

All test leads and testers will require Android phone and Android tablet.

|  |  |  |
| --- | --- | --- |
| **DeviceName** | **Version** | **Assigned To** |
| Mobile | Galaxy Nexus API 28 | Avni Sood |
| Mobile | Galaxy Nexus API 22 2 | Avni Sood |
| Mobile | Pixel XL API 28 | Avni Sood |
| Tablet | Nexus 10 API 28 | Avni Sood |

## Software Requirements

The standard items required for all testing efforts include the following items.

* <Android tools>
* <Appium>
* <IDEs>
* <Project management tool>
* <Selenium>
* <Nunit>

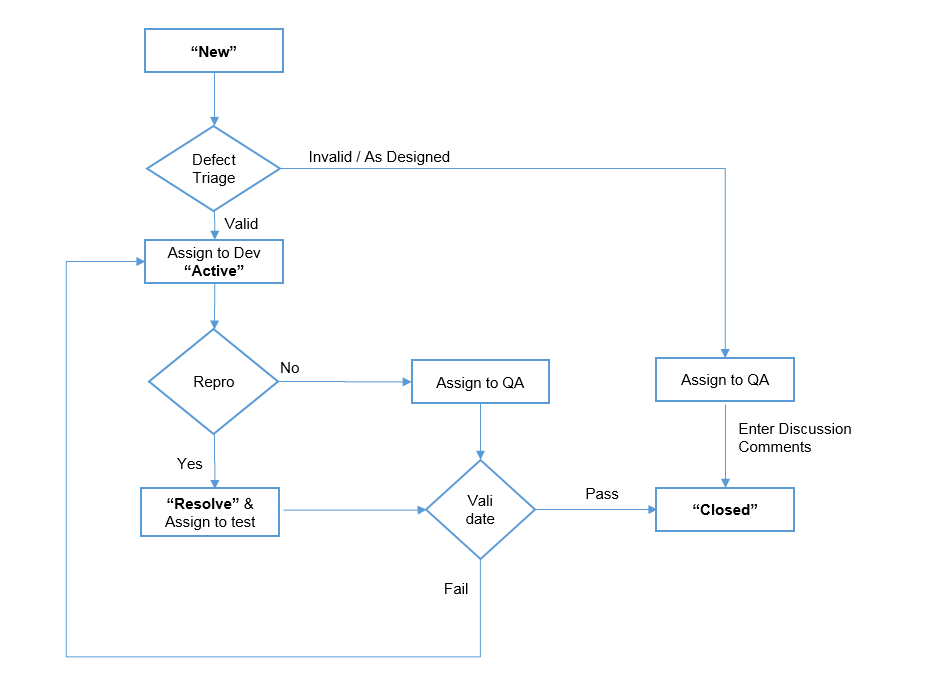
# List of Outstanding Issues

* The network service for actually placing the order isn't built yet.
* The spec doesn't call for any mechanism to distinguish an operator, who may both order things and edit the menu, from a customer, who may only order things

# Defect Management

All the defects are maintained and tracked with in VSO under its Project space. And the Process followed in managing it is detailed below

**Defect Process Flow**



***Defect Process Definition:***

* Once a defect is identified by Tester and created as ‘**NEW’** defect
* On Daily standups, Defects in NEW status are triaged
* During Defect Triage, Design Architect evaluates the validity of defect and decision made on its validity
* Valid – Assign to Developer
* Invalid/As Designed – Assign it to QE team
* On assigning the defect to Developer, status is set to ‘**Active’**
* On Reproducing the defect, Developer fix the defect and set it to ‘**Resolve’** status to validate the defect by QA team
* If the defect Passes, it will be ‘**closed’** by QE team
* If the defect fails, it is reassigned to Developer
* Note: If it is ‘As designed’ and acceptance criteria exist in user story, it will be communicated to Business Analyst and appropriate correction will be made with discussion notes

# Test Deliverables

|  |  |  |
| --- | --- | --- |
| Deliverable Name | Owner | Audience |
| *Test Cases* | *Avni Sood* | *Project Team* |
| *Test Case Review* | *XYZ* | *Project Team* |
| *Test Plan* | *Avni Sood* | *Project Team & Leadership* |
| *Exit Report* | *Avni Sood* | *Project Team & Leadership* |
| *Weekly Status Report* | *Avni Sood* | *Leadership* |
| *Daily Status Report* | *Avni Sood* | *Client* |

# Assumptions/Dependencies/Risk

## Assumptions

* Requirements, Test Scripts and Test Cases will be signed off by Business SMEs prior to start of Testing.
* Requirements available in VSTS will be the baseline requirements for development of test scenarios and detail level test cases.
* QE team will get required Test data prior to start of testing cycle.
* QE team will get testable build before the start of execution cycle.
* Number of critical/show stopper defects to be reported during all phases. It will have an impact on the execution timeline as the test execution will have dependencies on the fixes of the same.
* No change of business requirements will take place between Build and testing phases.
* All the user stories provided has necessary acceptance criteria for the QE to start working on analysis & design

## Dependencies

* APK file is available to all QE associates and in working condition
* Delivery of applications based on defined timelines
* Other teams available during the testing phase to ensure that their portion of testing is done as required
* Business SME support – for understanding business flows and business critical transactions in terms of user stories
* All third party application will be made available during all phase of testing
* Test data will be made available during all phase of testing

## Risk

* User stories without acceptance criteria will delay the analysis, development and test design activities of QE, which in turn will impact on schedule/timeline of deliverable
* Test devices are being shared across team, not having an independent device for QE team will delay the testing activities
* Non-Compliance to agree on plans and schedules.
* Non-availability of test device/environment may cause delay in delivery.
* Requirement/scope instability causing repetitive work.
* Test Data Instability and Unavailability will impact on schedule/timeline of deliverable
* Non-availability of access to data/application during testing phase
* Incomplete nonfunctional requirements/baselines.
* Dependencies on external applications, interfaces, or utilities not documented

# Entrance/Exit Criteria

## Entrance Criteria

Following is a list of documents and steps required for entrance to Testing:

* Project Plan -- Project Manager
* The Project Plan includes: (a) the point of contact; (b) team member roles, responsibilities, and contact information; (c) applications affected and their owners and contact names; (d) testing architecture; (e) dates and number of test iterations; (f) issue criticality definitions / resolution times; (g) testing logistics; and (h) list of approvers.
* Requirements Documentation
* Business Requirements – Business Analyst
* Functional Specifications – Business Analyst
* User Stories / Test Requirements with approval signatures – Business Analyst
* Architecture (if applicable, for big projects) – Architecture Team
* Code Unit and System Test is 100% complete and approved – Development Team
* Code is delivered with no known unresolved severity 1 or 2 defects – Development Team
* Android devices are ready for testing and handed over to test team.- Environment/Development Team

## Exit Criteria

Following is a list to be fulfilled for exit from Testing:

* Execution of all test cases are complete, i.e., 100% of cases have been executed and passed or deferred
* At least 90% of applicable test cases passed
* Zero priority 1 and severity 1 & 2 bugs and/or issues
* No priority 1 or severity 1 bugs and/or issues fixed in last day before Go-Live
* Project Teams sign off on technical documents
* Final Test Results and Test Exit Report have been submitted
* The functionality is ready for the next phase, i.e., UAT or Deployment
* Product owners sign off on functionality
* UAT Exit meeting conducted

# References

All the user stories & functional test cases referenced for evaluating the Snack Truck Application is available in below mentioned links:

***User Stories:***

User stories that are created are available in the following link <Link will be provided when available>

***Specification Document***:

Spec followed to prepare this plan <The Snack App-Homework.pdf>

***Test Execution Path:***

Test Execution results are being maintained in the following Path that are been executed as part of Snack Truck release <Link will be provided when available>

# Glossary

* + 1. Provide relevant terms and abbreviations.

|  |  |
| --- | --- |
| Acronym | Description |