**TEST PLAN FOR BLINKIT**

1. INTRODUCTION…………………………………………………………………………………………………………………….

1.1. Purpose……………………………………………………………………………………………………………………………

1.2. Project Overview……………………………………………………………………………………………………………..

2. TEST STRATEGY………………………………………………………………………………………………………………………

2.1. Test Objectives…………………………………………………………………………………………………………………

2.2. Test Approach………………………………………………………………………………………………………………..

2.3. Test Strategy………………………………………………………………………………………………………………….

3. TEST ENVIRONMENT DETAILS……………………………………………………………………………………………….

3.1. Test Environment Requirements……………………………………………………………………………………

3.2. Test Environment Setup....................................................................................................

3.3. Test Environment Sharing..................................................................................................

3.4. Test Environment Limitations and Mitigations. ................................................................

3.5. Test Environment Maintenance.. .......................................................................................

4. TEST DATA DETAILS…………………………………………………………………………………………………………………

4.1. Test Data Requirements... ..................................................................................................

4.2. Test Data Sharing.. .............................................................................................................

4.3. Test Data Collection/Generation.. .....................................................................................

4.4. Test Data Limitations and Mitigations………….....................................................................

4.5. Test Data Maintenance…………………………………………………………………………………………………

4.6. Backup (Test Environment and Test Data).. ………………………………………………………………

5. TRAINING…………………………………………………………………………………………………

5.1. Training Identified... …………………………………………………………………………………………………

5.2. Training Material and Approach. ……………………………………………………………………………………………

6. QA DELIVERABLES. …………………………………………………………………………………………………

7. QA TEAM-Roles and Allocations. ………………………………………………………………………………………………….

8. Test Environments and Type of Testing. …………………………………………………………………………………………

9. Test Scope……………………………………………………………………………………………

9.1. Included…………………………………………………………………………………………………

9.2. Excluded.. …………………………………………………………………………………………………

9.3. Possible Test Techniques. …………………………………………………………………………………………………

9.4. Testing Limitations & Workarounds……………………………………………………………………………………

10. QA SCHEDULE Releases. …………………………………………………………………………………………………

11. CROSS APPLICATION DEPENDENCY…………………………………………………………………………………………

12. DEPENDENCY with internal/External third party systems. …………………………………………………………

13. TEST Entrance and Exit Criteria. …………………………………………………………………………………………………

14. TEST Metrics.. ………………………………………………………………………………………………

15. Risk and Assumptions... …………………………………………………………………………………………………

16. Tools and Techniques... …………………………………………………………………………………………………

17. Decision Made…………………………………………………………………………………………………

18. Sign Off…………………………………………………………………………………………………

1. INTRODUCTION

1.1. Purpose

This test plan describes the testing approach and overall framework that will drive the testing of blinkit Application. The document introduces

* Test Strategy: rules the test will be based on, including the givens of the project (e.g: start/end dates. objectives, assumptions); description of the process to set up a valid test (eg: entry/exit criteria creation of test cases, specific tasks to perform, scheduling, data strategy).
* Execution Strategy: describes how the test will be performed and process to identify and report defects
* Test Management: process to handle the logistics of the test and all the events that come up during execution (eg: communications, escalation procedures, risk and mitigation, team roster)

1.2. Project Overview

* customer uses the bilnkit App for buy a grocery thing in online, it delivery with in minitue.
* Provide an online platform for businesses to buy their products and services
* Include features such as shopping carts, product listings, and search functionality
* Allow customers to easily purchase products or services online
* Provide a convenient shopping experience for customers
* Expand business reach and increase sales
* May include additional features such as customer reviews, ratings, and recommendations.

2. TEST STRATEGY

The objective of the test is to verify that the functionality of blinkit works according to the specification.

The test will execute and verify the test scripts, identify, fix and retest all high and medium severity defects per the entrance criteria, prioritize lower severity defects for future fixing via CR.

2.2. Test Approach

Testing will be planned and executed as functional black box testing, however, when necessary, the white box testing techniques will be applied

This approach will be reflected in the strategies for writing and executing the test cases.

Most of the testing will be executed manually.

2.3. Test Strategy

List the various types of tests to consider for the project

**Distributed QA testing** for some release, the responsibilities for quality control or tests would be distributed to various project team members (including development, DBA, business users, etc. For releases that depend on team members (in addition to QA) to plan and execute tests or quality control checks, this responsibility should be clearly defined among the team, in order to avoid gaps or misunderstanding the coverage for the shared responsibilities for quality.

**Functional**-System and Regression testing

**Non-functional** - Session management (time-outs, race conditions, multi-session support, such as logging in when another session is activel, performance (page load time, volume, stress), logging conditions, site switching, FS temporary down, STG, PROD advanced maintenance notifications tests,

**Security testing**-Burp suites, SQL Injections, Cross site vulnerability

**Automation Testing** (plan automation test for Smoke test, regression test, health check)

Perform **System testing** by using real mobile device or android/iOS emulators for Mobile Application.

**3rd Party Integration** - The project involve new integration with 3rd party system or components. GA Lead needs to take into account any communications or coordination for testing with the 3rd party system (proactively analyze and prepare for integration testing) and also communicate very clearly about the issue or testing supports/requirements to technical liaison/project manager and escalate if there is any delays or not receiving proper response from the third party vendor. QA team should understand about third party system design and different type of response (inputs/outputs parameter's) which will requires to use as inputs for CIA resource estimation, planning and allocation: CA lead should plan and communicate to QA director if there is any impact on the schedule due to resource shortage or schedule need to be revisited due to other high priority tasks .Mobile App Permissions QA lead should give more attention while testing web App .permissions settings or configurations and think end user perspective and test different way like uninstall/install and if any pop up comes with allow / deny or any permissions then we have to try different combination, and should include into regression test cases

QA lead should work with dev lead to get the impact analysis document.

QA lead should review the test cases, RTM, test results, bug reports and provide A release report along with Pre production QA assessment

QA lead work with UAT team and perform Pre UAT demo the UAT team

QA lead should review the test cases, along with Pre production QA assessment

QA had work with UAT team and perform Pre UAT demo the UAT team

QA lead work with director and provide the required metrics

QA had has to monitor of tasks and time logged into each of the assigned tasks and ensure that QA resource logged the time as per QA estimations

QA lead should inform to Dev lead if there is any changes requirements/08/ Technical) and there is no requirement for that changes in BRS/PRS then Dev lead work with PM to create a tasks/ticket for that and review with product team then re-assign to QA lead for testing if accepted by project team/product team approval.

3. TEST ENVIRONMENT DETAILS

This section will details the requirements, setup, limitations, maintenance for test environment required for effective testing of the project.

3.1.Test Environment Requirement

Elaborate the detailed requirements for the test environment, for examples

* Single or multiuser environment
* Single or multiplatform
* Compatibilities consideration
* Testing methodology to be applied and needed requirement for each testing

3.2. Test Environment Setup

Elaborate the actual setup of test environment, for example

* Hardware components configurations and setups
* Software components configurations and setup

3.3. Test Environment Sharing

If test environment are shared with other team or departments please consider the impact of sharing test environment and will sharing affect test results

3.4. Test Environment Limitations and Mitigations

This section will identify the limitations of test environment and what are to be done to mitigate these limitations in order to prevent it from affecting results,

3.5. Test Environment Maintenance

Test environment needs to be maintained to ensure it function in tip-top condition. Detailed the maintenance tasks and schedule, as well as roles and responsibilities if such maintenance tasks are perform outside of the software test team.

4. TEST DATA DETAILS

4.1.Test Data Requirements

* Type of test data needed Range of test data needed
* Range of test data needed
* test data Test data
* Relationship between different validity
* Test data

4.2. Test Data sharing

* Should test data be shared with other teams within the organization.

4.3. Test Data Collection/Generation

* Elaborate how test data are collected or created. Elaborate the creation methodologies / steps. Provide reference confluence pages if any for test data creation or steps to follow).

4.4. Test Data Limitations and Mitigations

* This section will identify the limitations of test data and what are to be done to mitigate these limitations in order to prevent it from affecting test results

4.5. Test Data Maintenance

* Test data needs to be maintained to ensure it remains relevant to testing requirements. Detailed the maintenance tasks and schedule, as well as roles and responsibilities if such maintenance tasks are perform outside of the software team

4.6. Backup (Test Environment and Test Data)

* Take a snapshot of the test environment, including the test data stored in it, using a backup and recovery software that is compatible with the platform.
* Create a backup of the test data itself by exporting it to a file or database that can be easily restored
* Store the backups in a secure location, such as an off-site server or cloud-based storage, to protect them from potential physical or cyber threats

5. TRAINING

5.1. Training Identified

The training identified for the Blinkit e-commerce app includes the following:

* User Interface and Navigation: Training on how to use the app's user interface and navigate through the various features and sections of the app.
* Product Management: Training on how to manage and update the products displayed on the app, including adding new products, editing existing products, and removing products that are no longer available.

5.2. Training Material and Approach

Training Material and Approach: The training material and approach for the Blinkit e-commerce app will depend on the specific needs and preferences of the target audience and the company. Some possible options include:

* Online tutorials and training videos: These can be accessed by users at any time and can be used as a reference for later use.
* In-person training sessions: This can be done by arranging training sessions for the employees or team members who will be using the app.

6. QA DELIVERABLES

|  |  |  |  |
| --- | --- | --- | --- |
| DELIVERABLE | LOCATION | AUTHOR | REVEIWER |
| Test Plan |  |  |  |
| Smoke test specification |  |  |  |
| Location Author/Contributor Reviewer |  |  |  |
| RTM |  |  |  |
| Test Specification |  |  |  |
| Bug reporting (JIRA)-Issue Log |  |  |  |
| Smoke test specification |  |  |  |
| Daily/Weekly Status rep |  |  |  |
| Core Regression tests QA check List |  |  |  |
| QA Release Report |  |  |  |
| Test Scenarios/Test Cases with test execution results |  |  |  |
| Pre UAT Test results UAT test results |  |  |  |
| QA Estimations |  |  |  |
| QA Metrics |  |  |  |

7. QA TEAM-Roles and Allocations.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| QA DIRECTOR | NAME | % OF TIME LOCATION | EMAIL | PHONE | RESPONSIBILITIES |
| QA MANAGER | NAME |  |  |  |  |
| QA LEAD | NAME |  |  |  |  |
| QA ENGINEER 1 | NAME |  |  |  |  |
| QA ENGINEER 2 | NAME |  |  |  |  |

8. Test Environments and Type of Testing.

|  |  |  |
| --- | --- | --- |
| TYPES OF TESTING | ENVIRONMENT | TESTING |
| Build Health check/Sanity check |  |  |
| Core Regression |  |  |
| Smoke testing Functional testing |  |  |
| Usability testing Database testing |  |  |
|  |  |  |
| Integration testing |  |  |
| Post deployment testing |  |  |
| Browser Compatibility testing |  |  |
| Pilot testing |  |  |
| System testing |  |  |
| UAT Testing |  |  |
| Backward compatibility testing. |  |  |
| Delivery acceptance testing |  |  |
| Regression testing |  |  |
| Performance testing |  |  |
| Automation testing |  |  |

9.Test scope

* It refers to the specific areas of the software product that will be tested, as well as the testing objectives and goals. The test scope for the blinkit app should include the following elements:

9.1. Included:

* The specific features and functionality of the blinkit app that will be tested. This might include the app's user interface, navigation, data entry and validation, search functionality, and any other features that are critical to the app's functionality.

9.2. Excluded:

* The areas of the blinkit app that will not be tested. This might include features that are not yet implemented, features that are considered low priority, or features that are known to have issues and will be addressed in a future release.

9.3. Possible Test Techniques:

* The different test techniques that will be used to test the blinkit app. This might include functional testing, usability testing, performance testing, and security testing.
* Boundary value analysis
* Main flow,alternate flow and exceptional flow
* Path coverage
* usability

9.4. Testing Limitations & Workarounds:

* Any limitations or constraints that may affect the testing process, such as time or budget constraints, or any workarounds that will be used to overcome these limitation.

10. QA SCHEDULE CAT Releases.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Planed date | Actual date | Milestone | Entry/exit criteria | owner | Additional info |
|  |  | Sprint 1 |  |  |  |
|  |  | Sprint 2 |  |  |  |
|  |  | Release candidate |  |  |  |
|  |  | Staging |  |  |  |
|  |  | Production |  |  |  |

11. CROSS APPLICATION DEPENDENCY

* Cross application dependency refers to the relationship between two or more software applications that are dependent on each other to function correctly. In the context of testing the blinkit app, cross application dependency refers to the dependencies that the blinkit app has on other software applications that are used in conjunction with it.
* if the blinkit app is an e-commerce platform, it may have a dependency on a payment gateway application, a shipping application, a inventory management application and other third party systems. Testing the blinkit app would need to consider the dependencies it has on these other applications and ensure that they are tested and working correctly.

12. DEPENDENCY with internal/External third party systems.

* Identify and document all external systems and services that the Blinkit app relies on for proper functioning
* Understand the interfaces and protocols used to communicate between the Blinkit app and these systems
* Test these interfaces and protocols thoroughly to ensure they are functioning correctly
* Monitor and test these external systems and services during the testing process to ensure they do not negatively impact the performance of the Blinkit app
* Have a plan in place for addressing any issues that may arise with these external systems during live deployment

13. TEST Entrance and Exit Criteria.

All requirements, design and development work for the app should be completed and signed off.

* A testable build of the app should be available for testing.
* All necessary test environments should be set up and configured.
* Test data should be prepared and made available for use.
* Test cases and test scripts should be developed and reviewed.
* Test team members should be trained on the app and testing procedures.

Test Exit Criteria for blinkit app:

* All test cases and test scripts should have been executed.
* All identified defects should have been reported and tracked to closure.
* Test metrics, such as test coverage and defect density, should be collected and analyzed.
* A final test report should be prepared and distributed to relevant stakeholders.
* The app should have been successfully tested and deemed ready for release by the
* test team and relevant stakeholders.
* The app should have met the acceptance criteria established in the requirements and design phases.

14. TEST Metrics.

TEST Metrics for blinkit ecommerce would include measurements such as:

* Website/app load time
* Number of successful transactions
* Customer satisfaction rates
* Number of bugs/errors found and resolved
* Time taken to resolve issues
* User engagement metrics (e.g. number of page views, time spent on site)
* Conversion rates (e.g. percentage of website visitors who make a purchase)
* A/B testing results (if applicable)
* SEO metrics such as keyword rankings and organic traffic.

15. Risk and Assumptions

Risk and assumptions for an blinkit ecommerce website may include:

* Credit card fraud or security breaches: There is a risk that sensitive customer information, such as credit card details, may be stolen during transactions or through hacking.
* Website downtime: There is a risk that the website may experience unexpected downtime, which can lead to lost sales and a poor customer experience.
* Inventory management: There is a risk that the website may run out of stock or overstock certain items, which can lead to lost sales or wasted resources.
* Shipping and logistics: There is a risk that items may be lost or damaged during shipping, or that shipping times may be delayed, leading to unhappy customers.
* Competitive pressure: There is a risk that the website may face intense competition from other ecommerce sites, leading to decreased sales and market share.
* Legal and regulatory compliance: There is a risk that the website may not comply with various legal and regulatory requirements, such as data protection laws or consumer protection laws, leading to fines or legal action.

Assumptions:

* The website is designed and developed according to the latest web standards and best practices.
* The website is compatible with the most popular web browsers and devices.
* The website is optimized for search engines.
* The website is able to handle large volumes of traffic and transactions.
* The website is able to integrate with third-party services, such as payment processors and shipping providers.
* Customers will be able to navigate and find products easily.
* Customers will be able to complete transactions smoothly.
* Customers will be satisfied with the products and services offered on the website.
* The website will be able to generate enough revenue to cover its costs and generate a profit.

16. Tools and Techniques

Tools and techniques for testing the Blinkit ecommerce app may include:

* Automated testing tools such as Selenium or Appium for functional testing of the user interface
* Performance testing tools such as Apache JMeter or LoadRunner for testing the app's response times and handling of high traffic
* Security testing tools such as OWASP ZAP or Nessus for identifying vulnerabilities and potential attack vectors
* Test management tools such as TestRail or Jira for tracking and reporting on test cases and results
* Code review tools such as SonarQube or Crucible for ensuring code quality and adherence to coding standards
* Continuous integration and deployment tools such as Jenkins or Travis CI for automating the testing and deployment process It's important to note that the specific tools and techniques used will depend on the project's specific requirements and constraints.

Top of Form

17.Decision Made

Decision Made for ecommerce would depend on the specific goals and objectives of the project and the testing phase. Some examples of decisions that may need to be made include:

* Choosing the appropriate test environment for the project, such as a staging or production environment

18.Sign Off

The test plan has been distributed to get sign from differnt department

* QA director
* Project manager
* Engineering manager
* Business analyst
* QA lead
* Engineering lead