





**Version History**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Change** | **Author** | **Approved by** |
| 1.0.0 | 09.19.2023 | Initiation of Log In, Dashboard, My Account, Manage Duplicate Records, Map, Record List, Add a Record. | Nigah Hossain | Abdul Quadir |
| 2.0.0 | 09.21.2023 | Update of Schedule | Nigah Hossain | Abdul Quadir |
| 3.0.0 | 09.22.2023 | Update of Reference, High Level Test Objective, Test Types | Nigah Hossain | Abdul Quadir |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Term** | **Abbreviation** | | | |
| UI | User Interface | | | |
| UX | User Experience | | | |
| OS | Operating System | | | |
| DRIVER | Data for Road Incident Visualization Evaluation and Reporting | | | |
| ARF | Accident Report Form | | | |
| RCF | Road Crash Form | | | |

**Abbreviation List**

**Table of contents**



[1. Introduction 4](#_gjdgxs)

[2. Reference 4](#_30j0zll)

[3. High Level Test Objective 5](#_1fob9te)

[4. Test Strategy 5](#_3znysh7)

[Strategy 5](#_2et92p0)

[Test Types 5-6](#_tyjcwt)

[5. Features to be Tested 6](#_3dy6vkm)

[6. Features not to be Tested 7](#_4d34og8)

[7. Test Estimation 7](#_2s8eyo1)

[8. Release Procedure 8](#_17dp8vu)

[9. Test Suspension Criteria 8](#_3rdcrjn)

[10. Test Acceptance Criteria: 9](#_26in1rg)

[11. QA Task List and Testing Process 9](#_lnxbz9)

[12. Test Environment 10](#_35nkun2)

[Hardware Requirement: 10](#_1ksv4uv)

[Software Requirement: 10](#_44sinio)

[Network Requirement: 10](#_2jxsxqh)

[Tools to be used: 10](#_z337ya)

[13. Schedule 11](#_3j2qqm3)

[14. QA Summary Report 11](#_1y810tw)

[15. Roles and Responsibilities 12](#_4i7ojhp)

[16. Risk and Contingencies 12](#_2xcytpi)

[17. Test Exit Criteria 13](#_1ci93xb)

[18. Bug Status Explanation 13](#_3whwml4)

[19. Test Deliverables 14](#_2bn6wsx)

[20. Test Plan Approvals 14](#_qsh70q)

# Introduction



The Dhaka Road Traffic Safety Project acknowledges this issue and seeks to transform the management of traffic and safety by introducing the DRIVER software - an advanced and versatile solution crafted to elevate road traffic safety, optimize traffic flow, and equip authorities with real-time data and insights. The DRIVER software embodies a fusion of cutting-edge technology, data analytics, and expertise in traffic management.

DRIVER, which stands for "Data for Road Incident Visualization, Evaluation, and Reporting," has been developed to gather and analyze data concerning incidents occurring in specific locations, such as traffic accidents or criminal activities. This initiative is a collaborative endeavor that unites traffic management specialists, software developers, and city officials. With the introduction of the DRIVER software, the project aims to inaugurate a new era of road safety and efficiency, serving as a model for sustainable traffic management practices. The path toward safer and more intelligent roadways commences with the DRIVER software—an embodiment of innovation, data-centric decision-making, and a dedication to enhancing the safety of Dhaka's streets for all its inhabitants and commuters.

This software offers a range of functionalities, including data input tools, graphical and mapping interfaces, customizable reporting and filtering tools, support for multiple concurrent users, and data export capabilities. The primary goal of the services is to deploy and configure the overall data collection and reporting aspects of the DRIVER software, enabling its utilization by the Dhaka Metropolitan Police (DMP) in their daily activities related to reporting traffic accidents and analyzing accident data.

# Reference



| **Ref. No** | **Document Title** |
| --- | --- |
| 1.0 | [[DRSP] Technical Document for Deployment and Configuration of the DRIVER Software V 1.0.0](https://drive.google.com/file/d/1Qt3zFSB7qIRpNSPi85ixydMEz8sev366/view) |
| 2.0 | [A Brief User Manual of the DRIVER System V 1.0.1](https://drive.google.com/file/d/1Au5o4EkEFl4p9SKt4iyPyxQZqalmPhO3/view?usp=drive_link) |
| 3.0 | [DRIVER\_Module list](https://docs.google.com/spreadsheets/d/1p7l_l9jO57jOGKwkPgrUofSOfoC0UMwm/edit?usp=drive_link&ouid=104381552986809267484&rtpof=true&sd=true) |
| 4.0 | [DRIVER\_RFP](https://drive.google.com/file/d/1Akm4RBpN5NsG2VS8fZ7WR9dzBSUsAe9u/view?usp=drive_link) |
| 5.0 | [QA-estimation-details\_Nigah Hossain](https://docs.google.com/spreadsheets/d/1OSQDvvuTSypwyeutcofDuTXal4rCyrRHEOElZj2Rt7w/edit?pli=1#gid=947949658) |

**Note:** The Project will be developed following a clone of Agile based methodology. Each Sprint duration will be 2 weeks. This test plan may also be changed according to the changes at any phase of testing.

# #High Level Test Objective



Based on the high-level goals of checking the DRIVER software for feature improvement, new feature implementation, issue fixing, and further maintenance work, the high-level test objectives can be outlined as follows:

* Ensure that all existing features of the application work according to the specified requirements. This includes testing against the functional requirements outlined in the project's Requirement Specification.
* Evaluate and validate any proposed feature improvements to ensure they meet the project's objectives and do not introduce new issues or regressions.
* Continuously identify and document software issues, including bugs, errors, and unexpected behavior. Prioritize and track these issues for resolution to improve the overall stability and reliability of the software with a focus on preventing the recurrence of known issues.
* Verify that the software's output and behavior align with the defined requirements. This ensures that the product meets the specific needs and expectations set forth in the Requirement Specification.
* Build trust and confidence with customers by delivering a product that meets their expectations and functions reliably. Customer satisfaction is a key goal, and this objective aims to achieve it through quality assurance.
* Ensure that the software complies with the detailed Requirement Specification, which serves as the basis for the project's goals and functionality.

# Test Strategy



## Strategy

To ensure the quality of the applications. testing will be conducted based on following approaches:

* **System Testing Strategy**
* Covers the entire software application or system, including all modules, components, and external interfaces, to ensure they work harmoniously together.
* Validates data inputs and outputs to ensure data integrity and accuracy.
* Ensures that integrated components work seamlessly and that data flows correctly between different parts of the system.
* **Functional Testing Strategies**
* Develop comprehensive test cases covering all functional requirements
* Verify that each functional point operates as expected.
* Include both positive and negative test scenarios.
* **UI/UX Testing Strategies:**
* Conduct UI/UX testing to evaluate the user interface and overall user experience of the software.
* Ensure that the application's UI meets client requirements and usability expectations.
* Assess the software's ease of use and accessibility.

## Test Level

Only the following test level will be conducted to ensure the quality:

* **System Testing:**

The system will be tested as a whole to ensure the consistency with each module

and features.

## Test Types

Following types of testing will be conducted to ensure the quality:

**Functional Testing:**

Functional testing will be conducted according to the software’s specifications considering the positive and negative scenario to ensure that all the functions and features work as intended according to the project's requirements and specifications.

**GUI Test:**

GUI testing focuses on the graphical user interface (UI) of the application. It ensures that the UI elements, such as buttons, menus, icons, and screens, are implemented correctly and that user interactions with the UI are smooth. It is to verify that the graphical user interface (GUI) elements of the project are visually consistent, interactive, and user-friendly.

# Features to be Tested.



| **Phases** | **Sprint** | **Features** |
| --- | --- | --- |
| 01 | 01 | 1. Complete UI/UX of the Log In 2. Login mechanism 3. Complete UI/UX for Dashboard 4. Dashboard mechanism 5. Complete UI/UX of the My Account 6. My Account mechanism 7. Complete UI/UX of the Manage Duplicate Records 8. Manage Duplicate Records mechanism 9. Complete UI/UX of the Map 10. Map mechanism 11. Complete UI/UX of the Record List 12. Record List mechanism 13. Complete UI/UX of the Add a record 14. Add a record mechanism |

# Features not to be Tested.



| **Phases** | **Features** |
| --- | --- |
| 01 | * Android/IOS * Database * Features- Incidents, Traffic Enforcement Assignments * Non functional Testing * Full Integration Testing * User Acceptance Testing * Performance Testing * Security Testing * Retesting * Full Regression Testing:   After issue fixing, full regression test requires much time and requires more resources, so we may not cover the full regression test. |
|
|
|
|

# Test Estimation



Accurate test estimation helps in resource allocation, scheduling, and project management. Testing effort may depend on several factors including.

* Test types and Test levels
* Quality of the Test basis
* Complexity of the problem domain
* Resource Requirements for documentation
* Time pressure
* Number of defects and the amount of rework required
* Test Case Count
* Test Execution Time
* Test Data Needs
* Risk and Contingency Planning

# Release Procedure



Releasing involves a structured process to ensure that the software meets quality standards and is ready for deployment. Below procedures will be followed for Release:

* Step-1: Requirement Analysis
* Step-2: Test Completion and Evaluation
* Step-3: Review Test Artifacts
* Step-4: Make internal release for QA on Sprint first day
* Step-5: Continue testing and report bug
* Step-6: Documentation Review
* Step-7: Make a Final Release for current Sprint on Sprint closing day
* Step-8: If no blocking issue is there and bug Severity is low, Application goes release otherwise release will not be done.
* Step-9: Remaining bugs will be fixed in next Sprint release
* Step-10: Continuous Improvement
* Step-11: Support and Maintenance

# Test Suspension Criteria



Test suspension criteria are predefined conditions or situations under which testing activities should be temporarily halted or suspended. These criteria help ensure that testing proceeds effectively and that any issues are addressed before continuing. Here are some typical test suspension criteria when Testing will be suspended, and QA team will reject the receivables upon the following criteria:

* **Blocking Issue Identification:**
* Testing will be suspended if a blocking issue is identified during testing. A blocking issue is typically a critical defect or problem that prevents further testing or renders the software unusable.
* The QA team will raise the issue to the respective stakeholders, drawing attention to the blocking issue and its impact on testing progress and software functionality.
* **Safety Risks:**
* If testing reveals safety-related risks or vulnerabilities that could pose a danger to users or the environment, testing should be suspended immediately to mitigate these risks.
* **Unfixed Bugs in Dedicated Release:**
* Testing will be suspended if a bug or issue identified during testing is not fixed in a dedicated release meant to address it.
* The QA team will verify whether the identified bug has been addressed in the dedicated release. If the issue persists, testing will not proceed until the bug is resolved.
* **Resource Constraints:**
* If there are significant resource constraints that hinder effective testing, such as a lack of test data, test environments, or testing tools, testing may be suspended until these resources are made available.
* **Release Without Release Note:**
* Testing will be suspended if a software release is made without accompanying release notes. Release notes are crucial for understanding the changes and updates in the software.
* The QA team will notify the stakeholders of the missing release notes and request that proper documentation be provided before testing can continue.
* **Data Integrity Issues:**
* If data integrity issues are identified that could compromise the accuracy or reliability of test results, testing should be suspended until data issues are resolved.
* **Environment Problems:**
* If the test environment encounters technical issues or is not properly configured, testing should be suspended until the environment is stable and suitable for testing.

**Note:** If any case will happen then QA have to raise issue to respective stakeholders

# Test Acceptance Criteria:

* Application UI should match as per provided UI specification.
* Application does not have any blocking issues.
* Application has 85% Test case coverage.
* Applications have covered supported required browsers (Windows Chrome).

# QA Task List and Testing Process



Below Tasks will be performed by the QA Team:

1. **Test Planning Phase:**

* Review project requirements and specifications.
* Identify Test areas.
* Define the scope and objectives of testing.
* Develop a comprehensive test plan.
* Establish testing goals and success criteria.
* Identify resources needed for testing (e.g., testers, test environments, tools).
* Determine the testing schedule and milestones.
* Document test entry and exit criteria.
* Define test documentation and reporting standards.
* Identify and document test risks and mitigation strategies.

1. **Test Design Phase:**

* Create detailed test cases and test scripts based on requirements.
* Develop test data and test scenarios.
* Design test environments and configurations.
* Review and validate test designs with stakeholders.

1. **Test Execution Phase:**

* Execute test cases and scripts according to the test plan.
* Log and report defects and issues.
* Monitor and track test progress and results.
* Verify that test environments are properly configured.
* Ensure test data accuracy and relevance.
* Document and report test execution status regularly.

1. **Defect Management and Resolution:**

* Prioritize and classify defects based on severity and impact.
* Maintain a defect repository for tracking and reporting.
* Ensure proper documentation of defect-related information.

1. **Test Reporting and Documentation:**

* Generate and distribute test summary reports.
* Document test coverage and traceability.
* Maintain test artifacts, including test cases and scripts.
* Deliver Test report.

1. **Release and Deployment:**

* Validate the release candidate build for production readiness.
* Verify that all test-related activities are complete.
* Collaborate with the release management team for deployment planning.
* Ensure proper documentation of release-related information.

1. **Continuous Improvement:**

* Scrum meeting
* Implement process improvements and best practices.
* Share insights and recommendations for future projects.

# Test Environment



Creating an appropriate test environment is crucial to conduct effective testing. To prepare the test bed forthe project followings are the requirement:

## Hardware and Infrastructure Requirement:

* Servers
* Workstations
* PC (Widows)

## Software and Operating Systems Requirement:

* **Operating System**: Windows 10 or above
* **Microsoft Office**
* **Google Chrome**

## Network Requirement:

* Internet connectivity to Laptop

## Tools to be used:

* **Test Case management**: Microsoft Excel
* **Document management**: NextCloud
* **Project management**: Redmine

Device oriented testing will be conducted as per following plan:

| Platform | Browser/OS | Device | Details | Comments |
| --- | --- | --- | --- | --- |
| Windows 10 Pro | Chrome  Version: 116.0.5845.188 | Laptop | Screen resolution: 1366 x 768 |  |

# Schedule



Schedule will be updated as Sprint feature release:

| **Phase** | **Feature Name** | No of Test  Items | Test Case Design | **Internal QA Release** | **Final Release** |
| --- | --- | --- | --- | --- | --- |
| 1 | Complete functional and UI/UX of Log In, Dashboard, My Account, Manage Duplicate Records, Map, Record List, Add a Record. | 7 | 169 | 09.25.2023 | 09.29.2023 |

# QA Summary Report

From this report all stockholders can view and judge the current project Quality

| **Sprint** | **URL** |
| --- | --- |
| 1st | <https://drive.google.com/drive/folders/1PLl8BUVy4gOgNcHChSDpJXIj2SgpKd1R?usp=drive_link> |

# Roles and Responsibilities



| **Resource Name** | **Responsibilities** |
| --- | --- |
| Abdul Quadir | Project Manager |
| Nigah Hossain | QA |

# Risk and Contingencies



**Schedule:**

**Risk:**

* Delays in the testing phase may impact the overall project timeline.

**Contingency:**

* Maintain a well-defined testing schedule with built-in buffers.
* Prioritize critical testing activities to ensure they are completed on time.
* Communicate regularly with project stakeholders to manage expectations.

**Testing:**

**Risk:**

* Incomplete or ineffective testing may lead to undiscovered defects or quality issues.

**Contingency:**

* Develop a comprehensive test strategy and plan that covers all critical areas.
* Implement test automation to expedite testing and increase coverage.
* Conduct thorough test case reviews and walkthroughs to identify gaps.
* Perform rigorous regression testing to catch unexpected side effects of code changes.

**Application Risk:**

**Risk:**

* Risks related to the specific features or functionalities.

**Contingency:**

* Identify and document specific application risks, such as potential vulnerabilities or areas prone to defects.
* Develop targeted test cases and scenarios to address these risks.
* Engage with subject matter experts to validate functionality.

**Performance Degradation under Load:**

**Risk:**

* High user load might lead to app slowdowns or crashes during peak usage times.

**Contingency:**

* Scale up server resources to handle the increased load.
* Notify users of potential performance issues and expected resolution time.
* Communicate with the development team to identify and address bottlenecks.

# Test Exit Criteria



The test exit criteria provide specific conditions that must be met before the testing phase can be considered complete Testing process of the DRIVER application will be ended if following criteria are met:

* All specified functions are functioning properly.
* All critical and high-severity bugs are identified, resolved, and retested.
* All test cases are executed and passed.
* Test coverage goals have been achieved for various aspects, such as requirements coverage, code coverage, and functional coverage.
* The software demonstrates stability and reliability during testing, with a low occurrence of critical crashes or issues.
* Testing is ongoing but if the PM requested to release the system.

# Bug Status Explanation



We maintain following status of the Bug in our Test Execution report:

**New:** The status of any defect or bug is logged in as To-do when it is discovered and verified. When a developer who has been assigned to the ticket is currently at work. Sets the Assignee to PM/TL/SE after creating the bug.

**Assigned:** PM/TL/SE changes the status to Assigned.

**In Progress:** Assignee changes the status to In Progress when s/he starts working. Assignee records Spent Time every day.

**Done:** This status denotes that the problem has been resolved, and further testing is permitted.

**Submitted:** Assignee changes the status to Submitted when s/he finished the task & set the Assignee to PM/TL. Assignee Record Spent time.

**Feedback:** If the review isn’t successful PM/TL changes the status to Feedback & Assignee to SE.

**Reviewed:** Assignee reviews and changes the status to Reviewed and Assignee to SQAE.

**Fixed but Failed:** Assignee (SQAE) retest and if not fixed then changes the status to Fixed but Failed and Assignee to PM/TL.

**Resolved:** Assignee (SQAE) retest and if fixed then changes the status to Resolved and Assignee to PM/TL.

**Verified:** The bug is retested by the tester after the developer fixes it. If no software flaw is found, the flaw has been addressed, and the status has been changed to "verified".

**Reopened:** The tester updates the state to "reopened" if the bug still exists after the developer has fixed it. The bug goes through its life cycle once more.

**Closed:** When a bug is declared to be fixed, the tester marks it as "Closed."

**Duplicate:** The status is changed to "duplicate" if the defect occurs twice or relates to the same notion as the bug.

**Rejected:** When a developer determines that a flaw is not actually present or the bug is invalid, the defect is changed to "rejected" by PM/TL & Assignee to SQAE.

**Deferred:** A bug is given the classification "Deferred" if it is not a top priority at the moment and is anticipated to be fixed in the upcoming version.

**Not a bug:** The status of a problem is "Not a bug" if it has no impact on the application's operation.

# Test Deliverables



Followings are the deliverables from QA for **DRIVER** project:

**Test Deliverables before Testing**

* Test Plan
* Test Cases

**Test Deliverables after Testing**

* Test execution report
* Test report (each sprint)
* Bug report

# Test Plan Approvals

| **Name** | **Roles** | **Signature** | **Date** |
| --- | --- | --- | --- |
| Abdul Quadir | Project Manager |  | 2023/09/19 |
| Abdul Quadir | Project Manager |  | 2023/09/21 |

