SFT221 NEE Group 1 – MS2

Test Plan

Member Present:

|  |  |
| --- | --- |
| **Student Name:** | **Student ID:** |
| Kim Ming Chau | 114320229 |
| Hak Kan Poon | 173333212 |
| Yuhong Fan | 129420220 |
| Elvin Karikari | 168318186 |
| Gordon Tan | 147206221 |

1. **Introduction**

**We are testing the functionality of a truck route management system. Specifically, the data structures of the header file of the system. The main objective is to ensure the system meets the specified requirements. The testing aims to identify any issues in the system and measure its performance, security, user acceptance etc.**

1. **Scope**

**What we will be testing:**

1. **System functionality: The correctness of the functions**
2. **Performance: How the system performs under different conditions. We will consider variables such as expected response and resource utilization.**
3. **Security: Vulnerabilities and weaknesses**
4. **Stress and Volume: Testing the performance and stability under extreme load conditions**
5. **Recovery: We will test to ensure its ability to recover from failures or disruptions**

**What we will not be testing:**

1. **Hardware dependences: Hardware such as GPS and Barcode Scanners will not be tested. Only focused on software functionality**
2. **Test Strategy**
   1. **3.1. System Test:** System testing will verify the overall system functionality, including all components and modules, to ensure they work together as expected.  
      **3.2. Performance Test:** Performance testing will be conducted to assess the system's response time, throughput, and resource utilization under different load conditions.  
      **3.3. Security Test:** Security testing will be performed to identify and address vulnerabilities and ensure the protection of sensitive data.  
      **3.4. Automated Test:** Automated tests will be developed to improve efficiency and effectiveness. These tests will be executed using suitable testing tools.  
      **3.5. Stress and Volume Test:** Stress testing will evaluate the system's performance and stability under extreme load conditions, while volume testing will assess its scalability and ability to handle large data volumes.  
      **3.6. Recovery Test:** Recovery testing will verify the system's ability to recover from failures, disruptions, or crashes and restore functionality.  
      **3.7. Documentation Test:** Documentation testing will review the accuracy, completeness, and clarity of user manuals, system documentation, and other related materials.  
      **3.8. Beta Test:** Beta testing will involve selected end-users testing the system in a real-world environment to provide feedback and validate its usability and functionality.  
      **3.9. User Acceptance Test:** User acceptance testing will involve end-users verifying that the system meets their requirements and is ready for deployment.
   2. 1. **how to understand requirements,**

The test design process begins with a thorough understanding of the requirements for the truck route management system. This involves analyzing the functional and non-functional requirements, user stories, and any other relevant documentation. The goal is to gain a clear understanding of the system's expected behavior, features, and performance.

* + 1. **build a traceability matrix,**

A traceability matrix is created to establish a clear link between the requirements and the corresponding test cases. Each requirement is mapped to one or more test cases to ensure comprehensive coverage. The traceability matrix helps in tracking the progress of testing and ensures that all requirements are adequately tested.

* + 1. **prepare test cases,**

Based on the identified requirements, test cases are prepared to verify the functionality of the truck route management system. Each test case represents a specific scenario or functionality that needs to be tested. For example, test cases can be created to validate functions such as adding a route, calculating distances, or finding the shortest path. The test cases include the necessary input data, expected results, and any preconditions or setup steps required.

* + 1. **and have them reviewed by another member of the quality assurance team.**

After preparing the test cases, they are reviewed by another member of the quality assurance team. This review ensures the accuracy, completeness, and effectiveness of the test cases. The reviewer examines the test cases for clarity, adherence to requirements, and adequacy of test coverage. Feedback and suggestions for improvement are provided to enhance the overall quality of the test cases.

1. **Environment Requirements**

**Hardware Requirements:**

* + 1. Test Computers: A set of dedicated test computers or workstations should be available to execute the tests. These machines should meet the minimum hardware specifications required by the truck route management system, such as CPU, memory, and storage capacity.
    2. Communication Equipment: Any communication equipment required for testing, such as network routers, switches, or modems, should be set up to ensure proper data transfer and connectivity between system components.

**Software Requirements:**

* + 1. OS: Could be either Windows, Linux, or Mac
    2. Test Harness or Testing Tools: Depending on the complexity and requirements of the truck route management system, a test harness or pre-existing testing tools may be needed to conduct the tests effectively. The test harness could be a custom-built solution, or an existing framework tailored to the system's needs.
    3. Continuous Integration: If the system utilizes a continuous integration process, a dedicated test computer or server should be available to automate the test execution and generate test reports. This ensures efficient and streamlined testing throughout the development lifecycle.

1. **Execution Strategy**
   1. During the testing cycle, all potential issues and bugs would be recorded at Jira. The defects would come with id, description, severity and reporter for ease for traceability.
   2. The criteria of completion of testing cycle is to get 95% of tests case pass. Moreover, there is no severe and critical defects.
   3. The defects would be categorized into below:

**Critical**

Critical defects have the highest level of severity as they cause complete failure or breakdown of the software.

**High**

Have a significant impact on the software's functionality, usability, or performance.

**Medium**

Bug which degrades the quality of a system but often has a work around to give the desired functionality

**Low**

Defects might be an unclear error message or some other minor error that has minimum impact on functionality

**Cosmetic**

Issue that makes the user interface less than optimal but still perfectly functional. For example, formatting, typo, etc

* 1. Team members need to go to Jira software to track his/her issues everyday and then update the status if there is any change. PM would remind team members if there is no progress for their issues. Would review the report together during the team meeting.

1. **Test Schedule**

****

1. **Control Procedures**
   1. Reviews – Team member reports the issue and confirm with Technical lead
   2. Bug Review Meetings – team members to agree the severity of the issues and suggested solution
   3. Change Request – after modification, responsible member to raise request for change of status to PM/Lead
   4. Defect Reporting – PM updates the status if agreed.
2. **Functions To Be Tested**

getNumRows() getNumCols() printMap()

addRoute() addPtToRoute() addPointToRouteIfNot()

addPointToRoute() getBlueRoute() getGreenRoute()

getYellowRoute() distance() shortestPath()

getPossibleMoves() getClosestPoint()

1. **Resources and Responsibilities**  
    9.1. Resources – Each member could use their own or school computer to do it in the Visual Studio, or even do it in GitHub  
    9.2. Responsibilities – Each member should help with editing, consulting, and setting up test cases
2. **Deliverables**

**When the milestone meets the finish date, then we will deliver any deliverables on GitHub.**

1. **Suspension / Exit Criteria**

**When there is no electric in the whole Toronto; therefore, no member could do the work**

1. **Resumption Criteria**

**When electric get back in Toronto; therefore, each member could work on the milestone again**

1. **Dependencies**  
    13.1 Personnel Dependencies

* Roles and responsibilities: Identify the personnel involved in the project, including delivery truck drivers, warehouse staff, and administrative personnel.
* Availability and scheduling: Determine the availability of personnel for testing, training, and project coordination to ensure smooth operations.

13.2 Software Dependencies

* Software requirements: Identify the software components required for the project, such as the delivery management system, route planning software, and tracking systems.
* Compatibility considerations: Ensure that the software components are compatible with the hardware and operating systems used by the company.

13.3 Hardware Dependencies

* Hardware requirements: Specify the hardware components necessary for the project, including delivery trucks, GPS devices, barcode scanners, and communication equipment.
* Configuration and setup: Ensure that the hardware components are properly configured, installed, and integrated with the software systems.

13.4 Test Data & Database

* Test data sources: Identify the sources of test data, such as customer information, delivery addresses, package details, and delivery routes.
* Database considerations: Specify the requirements for the database used to store and manage customer information, delivery status, and package tracking.

1. **Risks**  
    14.1. Schedule

* Schedule constraints: Identify potential risks related to project timelines, including delays in software development, equipment procurement, or resource availability.
* Mitigation strategies: Define strategies to address schedule-related risks, such as adjusting project timelines, prioritizing critical tasks, or allocating additional resources if needed.

14.2. Technical

* Technical risks: Identify potential risks associated with technical aspects of the project, such as integration challenges, software bugs, or hardware failures.
* Mitigation strategies: Specify strategies to mitigate technical risks, such as conducting thorough testing, implementing quality assurance processes, and having backup systems in place.

14.3. Management

* Management risks: Identify risks related to project management, including inadequate communication, scope creep, budget constraints, or resource allocation issues.
* Mitigation strategies: Define approaches to mitigate management-related risks, such as establishing effective communication channels, conducting regular project status meetings, and monitoring project scope.

14.4. Personnel

* Personnel risks: Identify risks associated with personnel, such as staff turnover, skill gaps, or dependencies on key team members.
* Mitigation strategies: Define strategies to mitigate personnel-related risks, such as cross-training team members, documenting processes and knowledge, and ensuring clear roles and responsibilities.

14.5 Requirements

* Requirements risks: Identify risks associated with unclear, incomplete, or changing requirements that may impact project delivery or customer satisfaction.
* Mitigation strategies: Define approaches to manage requirements-related risks, such as conducting thorough requirement analysis, maintaining clear communication with stakeholders, and implementing a change management process.

1. **Tools**

* Testing tools: Identify the tools required for testing, such as test management software, bug tracking systems, and performance testing tools.
* Selection and setup: Determine the appropriate tools based on project requirements, ensure compatibility with other systems, and establish proper configuration and setup.

1. **Documentation**

* Documentation requirements: Specify the types of documentation needed for the project, such as user manuals, system documentation, test scripts, and project reports.
* Documentation standards: Define the standards, templates, and guidelines for creating and maintaining project documentation to ensure consistency and clarity.

1. **Approvals**

* Approval process: Outline the process for obtaining approvals for the project plan, test plan, and other deliverables.
* Stakeholder involvement: Identify the key stakeholders responsible for reviewing and approving project documentation and deliverables.
* Criteria for approval: Define the criteria that must be met to obtain approval, such as adherence to industry standards, completion of testing milestones, and compliance with project requirements.