Test Plan Template

1. **Introduction**
   1. Test Plan Objectives
      1. **Our team is hired by a local delivery service company to design software to optimize delivery routes and improve overall delivery efficiency with limited resources. Through our testing efforts, we aim to make sure we meet all customer requirements, provide an easy-to-use user interface, and eliminate bugs in our program as much as possible. We hope to deliver a high-quality software at the end of our project deadline.**
2. **Scope**
   1. The scope of our testing is limited to functional testing and user acceptance testing, which will check if our software produces the expected output and whether we meet all business requirements. Stress testing and security testing will not be conducted.
3. **Test Strategy (Worksheet below)**



1. **Environment Requirements**

**a. We plan to conduct most of the testing on developer’s laptops. When necessary, Visual Studio Unit Test tool might be used which requires Windows operating system with at least 4GB RAM.**

1. **Execution Strategy**
   1. Developers will execute test cases according to the test plan. Test cases are then recorded with pass/fail status ensuring project requirements are met and bugs or unwanted behaviors are documented. These test cases will do a variety of types of testing but mainly will perform functional, integration, regression, and performance testing.
   2. Entry criteria of test should have project in a state where it successfully builds. Exit criteria of test is based on whether it passes.
   3. Severity of defects, bugs and unwanted behavior can be classified as one of five states:
      1. **Critical** - Causes the system to crash or produce anomalous results.
      2. **High -** Causes lack of program functionality and might have a work around.
      3. **Medium -** Bug which degrades the quality of a system but often has a workaround to give the desired functionality.
      4. **Low** - Might be an unclear error message or some other minor error that has minimum impact on functionality.
      5. **Cosmetic** - Something that makes the user interface less than optimal but still perfectly functional.
   4. **Test Reporting**
      1. Test report cases in an excel spreadsheet depicting the test cases as well as their results and type of tests.
      2. These reports should be produced for every function in the process of being implemented so members and group leader can review before integrating into production.
      3. Group leader to act as quality assurance team to review test cases and find potential bugs to assign back to developer for revision.
2. **Test Schedule**

|  |  |  |
| --- | --- | --- |
| **Item** | **Est. Timeline** | **Details** |
| **Test planning** | **1 Week** | **Produce a test plan report** |
| **Test cases designing** | **1 Week** | **Design the specific test cases for functions/modules, as well as integration tests.** |
| Test execution | **1 Week** | **Execute tests, make records of tests and fix issues/bugs** |
| **User acceptance test** | **1 Week** | **Test for business requirements** |
| **Final review** | **1 Week** | **Review the software & prepare for delivery** |

1. **Control Procedures**

a. Reviews: developers to conduct code reviews periodically, when necessary, team will meet to discuss major changes or updates.

b. Bug Review Meetings: Allocate 15-30 minutes during the weekly team meeting to discuss major issues and feasible solutions as a team. One of the developers will be tasked to implement the discussed solution and update the team if any issues come up. Bugs are also prioritized based on severity and impact on the project.

c. Change Request: Change requests are documented and submitted using Jira. Team lead or the team together will evaluate the impact of changes on project timeline, and either approve or reject changes.

d. Defect Reporting: Defects should be timely reported using standardized format and reported on Jira. A developer will be assigned to the issue to fix defects, who is also responsible to update the team of the fix progress. Team lead is responsible to keep track of progress of all defects and ensure timely delivery.

1. **Functions To Be Tested**

Based on the required output examples, test and modify existing codes to make it meet the project output requirements. Here are some functions that may need to be added and tested.

//validate weight of the shipment

int isValidWeight(double weight);

//validate box size

int isValidSize(double size);

// Validate input destination

int isValidDestination(const char\* des);

// Determines the appropriate route based on the destination, weight, and size of the shipment.

struct Route shipmentRoute(const char\* des, double weight, double size);

// Converts a destination to a point on the map.

struct Point desToPoint(const char\* des);

//display final route

void display(const struct Route\* route);

1. **Resources and Responsibilities**  
   Assign clear responsibilities to team members for different testing activities such as test case creation, execution, bug tracking, testing and documentation. This organization helps ensure efficient progress throughout the testing process.
2. **Deliverables**

Clearly define each deliverable should have a specified format and deadline, such as list each person's assigned project tasks, reflecting its status, and expected completion time on Jira. It should be clear who is responsible for producing and reviewing each one. Teams are expected to collaborate effectively by utilizing a Git repository for version control of all work. We are encouraged to actively participate in weekly meetings, select tasks from the Kanban board, and take responsibility for a balanced contribution to the project workload.

1. **Suspension / Exit Criteria**

Establish clear criteria for temporarily halting the testing process or for concluding it. Suspension criteria could include severe defects that block further testing, while exit criteria could involve reaching a certain test coverage threshold or resolving all critical bugs. This ensures that testing is only concluded when the software meets all quality standards.

1. **Resumption Criteria**

Define the conditions that must be satisfied to resume testing if it has been suspended. This may involve fixing critical bugs, acquiring additional resources, or meeting pre-defined testing prerequisites. Clarifying these criteria helps in efficiently managing delays and ensuring that testing resumes at an appropriate time.

1. **Dependencies**  
    12.1 Personnel Dependencies

**Project specifications from professor**

12.2 Software Dependencies

**Visual Studio**

12.3 Hardware Dependencies

Access to lab computer or personal laptop/computer  
12.3 Test Data & Database

Test data is dependent upon test cases being documented.

1. **Risks**

Schedule:

* + 1. Description: From time to time there might be potential delays caused by unforeseen circumstances such as sickness, weather, etc.
    2. Mitigation: Each team member should timely communicate with the team if any issue that might cause delay to their work. Team should also Build contingency time into the schedule to account for unexpected delays. If severe delay is expected, team should discuss with stakeholders and come up with feasible solution.

Technical:

1. Description: Unforeseen technical challenges or complexities in implementing certain features.
2. Mitigation: Read through specifications of project carefully and foresee potential issues if possible. Discuss with key stakeholder if technical issue creates major roadblock for the development efforts.

Management:

1. Description: Ineffective project management might lead to miscommunication or conflicting priorities. Or lack of clarity in roles and responsibilities among team members.
2. Mitigation: Establish clear communications between team members; delegate specific, feasible, measurable tasks during team meeting to each team member; use Jira to keep track of progress. Conduct regular team meetings to ensure alignment on project goals and objectives.

Personnel:

1. Description: Personnel changes such as a team member leaving the team can cause delays to the project.
2. Mitigation: Establish clear communications within the team. Team members should let the team know as early as possible if unforeseen circumstances occur. Have open communication with stakeholders regarding impacts on the project.

Requirements:

1. Description: If requirements become unclear or difficult to implement, team will communicate such situations in a timely manner and ask for clarifications.
2. **Tools**

a. Visual Studio (main development tool)

i. Debugger (bug fix)

ii. Unit Test Feature (testing)

b. Git, GitHub (project delivery)

c. Jira (progress tracking, issue reporting, task delegation)

d. MS Teams (communication)

**16. Documentation**

**a. In order to keep a systematic record of our project, and organize the information efficiently, team agrees to maintain good documentations of team meetings (scrum report), major development milestones reached, bug fixes, testing efforts (test strategy plan and test execution).**

**17. Approvals**

**a. Review and approvals of documentations, development decisions and directions, final project delivery by key stakeholders, which in our case is the course instructor.**