



METAOPTION

Sample Test Plan on InTownLive
[By: Testing Team METAOPTION]

Introduction

General

This document describes the approach used in order to validate and verify the functionality of the InTownLive application. Test cases will be created and used for the regression testing, effort on each new release or patch to the application. This will continually ensure that no new defects will be introduced into the software and ensure a repeatable, consistent test effort.

As this document has been produced early in the testing phase, it does not present specifics of the testing effort, but a rather general view of the type of testing carried out.

InTownLive application Overview

This project is developed for InTownLive U.S.A. InTownLive is a premium web based tourism guide which will help users to explore America's most charming towns. The main website provides town description, photo gallery, slideshows, 2D panorama images of town, business names & location on map etc. It has a business listing / advertisement module for customers willing to display their organizations listing / advertisement in different towns and categories. There are both free listing and register and paid (with advertisement) options for the user.

The purpose of this test plan is to identify the specific efforts required to test the InTownLive application for regression testing purposes as well as detail the proposed test schedule.

Scope

The test development effort will confirm that:

The tests meet all the business functionality requirements for the application.

No new problems will be introduced into each new build.

The test effort is sufficiently documented in the form of Test Cases in order to ensure repeatable, consistent tests.

Defects and requirements are tracked using Project Management & Bug Tracking software.

Test Strategy

Overview

Testing will be done using a structured approach in order to validate the functionality of the application using test cases developed in a regression test environment. Test Specification allowed the test team to produce documents and reports of all test cases developed and executed in real-time.

Categories of Testing

Integration

Business functionality

Regression

Performance etc

Integration Testing

Integration testing is a systematic approach for constructing the program structure while at the same time tests to uncover errors associated with interfacing. Integration testing will verify that the software units operate properly as a unit.

Business Functionality Testing

Business functionality testing will be the majority of the workload. This is where all areas of the application will be exercised from the perspective of the end user.

The objective of function test is to measure the quality of the functional (business) components of the system. Tests verify that the system behaves correctly from the user / business perspective and functions according to the requirements,

models or any other design paradigm used to specify the application. The function test must determine if each component or business event: performs in accordance to the specifications, responds correctly to all conditions that may be presented by incoming events / data, moves data correctly from one business event to the next (including data stores), and that business events are initiated in the order required to meet the business objectives of the system.

Regression Testing

Each time a new module is added as a part of integration, the software changes. New data flow paths are established, new I/O may occur and new control logic is invoked. Regression testing will be conducted after each new build of the software is released and will ensure that no new defects will be introduced into the software.

Performance Testing

Performance testing is the process of identifying the speed or effectiveness of and computer, software program or device. Performance testing can verify that an application meets the specifications claimed by its manufacturer or vendor. This process can involve tests in lab, such as measuring the response time or the number of MIPS (millions instructions per second) at which a system functions. Qualitative attributes such as scalability, interoperability may also be evaluated. Performance testing is done in conjunction with the stress testing.

In Performance we can compare two or more devices or programs in terms of parameters such as speed, data transfer rate, bandwidth, throughput, efficiency or reliability.

Slow data transfer rate may be intrinsic in hardware but can also result from software-related problems, such as:

- Too many applications running at the same time
- A corrupted file in a Web browser
- A security exploit

- Heavy-handed antivirus software
- The existence of active malware on the hard disk.

Work Flow and Defects Tracking

This section defines the interface between the Test team and the Development team, and provides definitions for some defect attributes.

Testing Work Flow

All testing will be carried out in the development environment.

The application will be tested and defects will be recorded in "Bug Track System". Defects will be sent to the development team leader. As the defects are fixed, the report will be updated by the developer and re-assigned to the test team for validation. The repairs to the software will be validated by the test team when the new software is tested. Once a repair will be validated, the defect will be closed. Should the repair be incomplete or not effective, it will be rejected, and the defect will be re-assigned to the developer to be fixed again. This cycle will be repeated until the defect has been properly repaired and validated.

Definitions applicable to defects

Defects generated during testing will be assigned a priority and a status level. These assignments can be changed as required if it is determined that an assignment is inappropriate.

Status

Open: A defect has been identified and opened by a test team member. Open defects will be assigned to a developer for repair through BTS (Bug Tracking System).

Closed: A potential defect will be determined not to be as a defect and will then be closed, or a defect will be repaired and validated.

Assigned: Bug assigned to the developers along with severity and the number of days required fixing it.

Fixed: Bug once fixed will again be retested under the same situation under which it has been generated.

Deferred: A defect will be recognized but won't be resolved for various reasons. A specific decision has to be recorded in Add Note/ Modify Status to leave a defect unresolved.

Severity

Crash: When application crashes by any means.

Major: Includes that issue which need to fix very soon.

Minor: Issues which does not influence the current functionality of the application.

Tweak: Some minor issues like spelling mistake, indentations etc.

Text: Some text formatting issues on any page.

Feature: New added features in the existing system.

Priorities

None: If the priority is general and does not require immediate action it is kept as none.

Urgent/Immediate: The defect prevents further testing of the function and must be resolved immediately to avoid test schedule delays.

High: The defect relates to an important function of the application, or one that will be used frequently, and should be resolved as soon as possible.

Medium: The defect will be resolved in the normal course of events. Most defects will be in this category.

Normal: The defect is not major and will be resolved as time allows.

Low: The defect is not major and may or may not be resolved. This type of defect does not affect the functionality of the application.

Reporting Any person, who has access to "Bug Tracking System", can generate a defect report using real-time data as test cases are executed. It is the responsibility of the Test team to

enter defects into the BTS (Bug racking System) as they are found in the software, and monitor their progress through the life cycle.

Completion Criteria for testing

Testing will be considered complete when:

- All test cases will be developing for regression purposes
- All Emergency and High priority defects identified will be closed
- Most, or all, Medium priority defects will be closed and
- Low priority defects and Enhancement requests may still not be closed. However, the clients are satisfied that they may use the application in its current state.

Schedule

Number of Resources	Activity	Approx. days per Resource	Responsibility
Enter # of resources required to complete the activity	Enter testing activity here. E.g.: Test case creation	Enter number of days estimate here	Enter name of proposed QA resource here