|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Plan for Nbyula**    **1.****Objectives**  **1.1 Purpose**  This document describes the test plan for test activities for a new marketplace website ‘Nbyula’ which just started development, for the upcoming 6 months until delivery. This Test Plan document supports the following objectives:   * + Identify existing project information and the software that should be tested.   + List the recommended test requirements.   + Recommend and describe the testing strategies to be employed.   + Identify the required resources and provide an estimate of the test efforts.   + List the deliverable elements of the test activities.   **1.2****Scope**  This Test Plan describes the integration and system tests that will be conducted on the website Nbyula.  It is critical that all system and subsystem interfaces be tested as well as system performance.  The interfaces between the following subsystems will be tested:   * 1. Registration   2. Finance System   3. Catalog   The external interfaces to the following devices will be tested:   * 1. Local PCs   2. Remote PCs.   Some of the performance measures to test are:   * 1. Response time for remote login to the registration system.   2. Response time to access the Finance System.   3. Response time to access the Catalog Subsystem.   4. User response time when system loaded with 200 logged in users.   5. User response time when 50 simultaneous accesses to the Catalog database.   **2.****Requirements for Test**  The listing below identifies those items (use cases, functional requirements, non-functional requirements) that have been identified as targets for testing.  **2.1 Data and Database Integrity Testing**  Verify access to Catalog Database.  Verify simultaneous record read accesses.  Verify lockout during Catalog updates.  Verify correct retrieval of update of database data.  **2.2. Function Testing**  Supplementary Specification: "The system shall interface with the existing Course Catalog Database System."  Supplementary Specification: "The system shall interface with the existing Billing System"  Supplementary Specification: "The client component of the system shall operate on any personal computer with a 486 Microprocessor or greater."  Supplementary Specification: "The system shall integrate with existing legacy system (catalog database)."  Supplementary Specification: "The system shall integrate with the existing Billing System."  **2.3 User Interface Testing**  Verify ease of navigation through a sample set of screens.  Verify sample screens conform to GUI standards.  Supplementary Specification: "The desktop user-interface shall be Windows 95/98 compliant."  Supplementary Specification: "The user interface shall be designed for ease-of-use and shall be appropriate for a computer-literate user community with no additional training on the System."  **2.4 Performance Testing**  Verify response time to access external Finance system.  Verify response time to access external Catalog subsystem.  Verify response time for remote login.  Verify response time for remote submittal of registration.  Supplementary Specification: "The system shall provide access to the legacy Catalog Database with no more than a 10 second latency."  **2.5 Load Testing**  Verify system response when loaded with 200 logged on users.  Verify system response when 50 simultaneous user accesses to the Catalog.  **2.6 Volume Testing**  None.  **2.7 Security and Access Control Testing**  Verify Logon from a local PC.  Verify Logon from a remote PC.  Verify Logon security through user name and password mechanisms.  **2.8 Failover / Recovery Testing**  None.  **2.9 Configuration Testing**  Supplementary Specification: "The client component of the system shall run on Windows 95, Windows 98, and Microsoft Windows NT."  Supplementary Specification: "The web-based interface for the C-Registration System shall run in Netscape 4.04 and Internet Explorer 4.0 browsers."  Supplementary Specification: "The web-based interface shall be compatible with the Java 1.1 VM runtime environment."  **2.10 Installation Testing**  None.  **3.****Test Strategy**  The Test Strategy presents the recommended approach to the testing of the software applications. The previous section on Test Requirements described *what* will be tested; this describes *how* it will be tested.  The main considerations for the test strategy are the techniques to be used and the criterion for knowing when the testing is completed.  In addition to the considerations provided for each test below, testing should only be executed using known, controlled databases, in secured environments.  The following test strategy is generic in nature.  **3.1****Testing Types**  *3.1.1**Data and Database Integrity Testing*  The databases and the database processes should be tested as separate systems. These systems should be tested without the applications (as the interface to the data). Additional research into the DBMS needs to be performed to identify the tools / techniques that may exist to support the testing identified below.     |  |  | | --- | --- | | Test Objective: | Ensure Database access methods and processes function properly and without data corruption. | | Technique: | * Invoke each database access method and process, seeding each with valid and invalid data (or requests for data). * Inspect the database to ensure the data has been populated as intended, all database events occurred properly, or review the returned data to ensure that the correct data was retrieved (for the correct reasons) | | Completion Criteria: | All database access methods and processes function as designed and without any data corruption. | | Special Considerations: | * Testing may require a DBMS development environment or drivers to enter or modify data directly in the databases. * Processes should be invoked manually. * Small or minimally sized databases (limited number of records) should be used to increase the visibility of any non-acceptable events. |       *3.1.2 Function Testing*  Testing of the application should focus on any target requirements that can be traced directly to use cases (or business functions), and business rules. The goals of these tests are to verify proper data acceptance, processing, and retrieval, and the appropriate implementation of the business rules. This type of testing is based upon black box techniques, that is, verifying the application (and its internal processes) by interacting with the application via the GUI and analyzing the output (results). Identified below is an outline of the testing recommended for each application:     |  |  | | --- | --- | | Test Objective: | Ensure proper application navigation, data entry, processing, and retrieval. | | Technique: | * Execute each use case, use case flow, or function, using valid and invalid data, to verify the following: * The expected results occur when valid data is used. * The appropriate error / warning messages are displayed when invalid data is used. * Each business rule is properly applied. | | Completion Criteria: | * All planned tests have been executed. * All identified defects have been addressed. | | Special Considerations: | * Access to the existing Catalog System and Billing System is required to run some of the identified System Tests on the Prototype. |     *3.1.3**User Interface Testing*  User Interface testing verifies a user’s interaction with the software. The goal of UI Testing is to ensure that the User Interface provides the user with the appropriate access and navigation through the functions of the applications. In addition, UI Testing ensures that the objects within the UI function as expected and conform to corporate or industry standards.     |  |  | | --- | --- | | Test Objective: | Verify the following:   * Navigation through the application properly reflects business functions and requirements, including window to window, field to field, and use of access methods (tab keys, mouse movements, accelerator keys) * Window objects and characteristics, such as menus, size, position, state, and focus conform to standards. | | Technique: | * Create / modify tests for each window to verify proper navigation and object states for each application window and objects. | | Completion Criteria: | Each window successfully verified to remain consistent with benchmark version or within acceptable standard | | Special Considerations: | * Not all properties for custom and third-party objects can be accessed. |     *3.1.4**Performance Profiling*  Performance testing measures response times, transaction rates, and other time sensitive requirements. The goal of Performance testing is to verify and validate the performance requirements have been achieved. Performance testing is usually executed several times, each using a different "background load" on the system. The initial test should be performed with a "nominal" load, similar to the normal load experienced (or anticipated) on the target system. A second performance test is run using a peak load.  Additionally, Performance tests can be used to profile and tune a system’s performance as a function of conditions such as workload or hardware configurations.   |  |  | | --- | --- | | Test Objective: | Validate System Response time for designated transactions or business functions under the following two conditions:  - normal anticipated volume  - anticipated worse case volume | | Technique: | * Use Test Scripts developed for Business Model Testing (System Testing). * Modify data files (to increase the number of transactions) or modify scripts to increase the number of iterations each transaction occurs. * Scripts should be run on one machine (best case to benchmark single user, single transaction) and be repeated with multiple clients (virtual or actual, *see special considerations below).* | | Completion Criteria: | * Single Transaction / single user: Successful completion of the test scripts without any failures and within the expected / required time allocation (per transaction) * Multiple transactions / multiple users: Successful completion of the test scripts without any failures and within acceptable time allocation. | | Special considerations: | * Comprehensive performance testing includes having a "background" load on the server. There are several methods that can be used to perform this, including:   + "Drive transactions" directly to the server, usually in the form of SQL calls.   + Create "virtual" user load to simulate many (usually several hundred) clients. Remote Terminal Emulation tools are used to accomplish this load. This technique can also be used to load the network with "traffic."   + Use multiple physical clients, each running test scripts to place a load on the system. * Performance testing should be performed on a dedicated machine or at a dedicated time. This permits full control and accurate measurement. * The databases used for Performance testing should be either actual size, or scaled equally. |     *3.1.5**Load Testing*  Load testing measures subjects the system-under-test to varying workloads to evaluate the system’s ability to continue to function properly under these different workloads. The goal of load testing is to determine and ensure that the system functions properly beyond the expected maximum workload. Additionally, load testing evaluates the performance characteristics (response times, transaction rates, and other time sensitive issues).   |  |  | | --- | --- | | Test Objective: | Verify System Response time for designated transactions or business cases under varying workload conditions. | | Technique: | * Use tests developed for Business Cycle Testing. * Modify data files (to increase the number of transactions) or the tests to increase the number of times each transaction occurs. | | Completion Criteria: | * Multiple transactions / multiple users: Successful completion of the tests without any failures and within acceptable time allocation. | | Special Considerations: | * Load testing should be performed on a dedicated machine or at a dedicated time. This permits full control and accurate measurement. * The databases used for load testing should be either actual size, or scaled equally. |     *3.1.6**Volume Testing*  This section is not applicable to test.  *3.1.7 Security and Access Control Testing*  Security and Access Control Testing focus on two key areas of security:  - Application security, including access to the Data or Business Functions, and - System Security, including logging into / remote access to the system.  Application security ensures that, based upon the desired security, users are restricted to specific functions or are limited in the data that is available to them. For example, everyone may be permitted to enter data and create new accounts, but only managers can delete them. If there is security at the data level, testing ensures that user "type" one can see all customer information, including financial data, however, user two only sees the demographic data for the same client.  System security ensures that only those users granted access to the system are capable of accessing the applications and only through the appropriate gateways.     |  |  | | --- | --- | | Test Objective: | Function / Data Security: Verify that user can access only those functions / data for which their user type is provided permissions.  System Security: Verify that only those users with access to the system and application(s) are permitted to access them. | | Technique: | * Function / Data Security: Identify and list each user type and the functions / data each type has permissions for. * Create tests for each user type and verify permission by creating transactions specific to each user type. * Modify user type and re-run tests for same users. In each case verify those additional functions / data are correctly available or denied. * System Access (see special considerations below) | | Completion Criteria: | For each known user type the appropriate function / data are available and all transactions function as expected and run-in prior Application Function tests | | Special Considerations: | * Access to the system must be reviewed / discussed with the appropriate network or systems administrator. This testing may not be required as it may be a function of network or systems administration. |     *3.1.8 Failover and Recovery Testing*  This section is not applicable to test.  *3.1.9 Configuration Testing*  Configuration testing verifies operation of the software on different software and hardware configurations. In most production environments, the particular hardware specifications for the client workstations, network connections and database servers vary. Client workstations may have different software loaded (e.g. applications, drivers, etc.) and at any one time many different combinations may be active and using different resources.     |  |  | | --- | --- | | Test Objective: | Validate and verify that the client Applications function properly on the prescribed client workstations. | | Technique: | * Use Integration and System Test scripts * Open / close various PC applications, either as part of the test or prior to the start of the test. * Execute selected transactions to simulate user activities into and out of various PC applications. * Repeat the above process, minimizing the available conventional memory on the client. | | Completion Criteria: | For each combination of the Prototype and PC application, transactions are successfully completed without failure. | | Special Considerations: | * What PC Applications are available, accessible on the clients? * What applications are typically used? * What data are the applications running (i.e. large spreadsheet opened in Excel, 100 page document in Word). * The entire systems, network servers, databases, etc. should also be documented as part of this test. |     *3.1.10**Installation Testing*  This section is not applicable to test.  **3.2****Tools**  The following tools will be employed for testing:   |  |  | | --- | --- | |  | **Tool** | | Test Management | Rational RequisitePro  Rational Unified Process | | Test Design | Rational Rose | | Defect Tracking | Rational ClearQuest | | Functional Testing | Rational Robot | | Performance Testing | Rational Visual Quantify | | Test Coverage Monitor or Profiler | Rational Visual PureCoverage | | Other Test Tools | Rational Purify  Rational TestFactory | | Project Management | Microsoft Project  Microsoft Word  Microsoft Excel | | DBMS tools | TBD |     **4.****Resources**  This section presents the recommended resources for testing, their main responsibilities, and their knowledge or skill set.  **4.1****Roles**  This table shows the staffing assumptions for the test of the Prototype.     |  |  |  | | --- | --- | --- | | **Human Resources** | | | | **Role** | **Minimum Resources Recommended**  (Number of workers allocated full-time) | **Specific Responsibilities/Comments** | | Test Manager | 1 | Provides management oversight  Responsibilities:   * Provide technical direction * Acquire appropriate resources * Management reporting | | Test Designer | 2 | Identifies, prioritizes, and implements test cases  Responsibilities:   * Generate test plan * Generate Test Suite * Evaluate effectiveness of test effort | | System Tester | 1 | Executes the tests  Responsibilities:   * Execute tests * Log results * Recover from errors * Document defects | | Test System Administrator | 1 | Ensures test environment and assets are managed and maintained.  Responsibilities:   * Administer test management system * Install / manage worker access to test systems | | Database Administration / Database Manager | 1 | Ensures test data (database) environment and assets are managed and maintained.  Responsibilities:   * Administer test data (database) | | Designer | 1 | Identifies and defines the operations, attributes, and associations of the test classes  Responsibilities:   * Identifies and defines the test class(es) * Identifies and defines the test packages | | Implementer | 1 | Implements and unit tests the test classes and test packages  Responsibilities:   * Creates the test classes and packages implemented in the Test Suite. |       **5.****Project Milestones**  Testing of the Prototype incorporates test activities for each of the test efforts identified in the previous sections. Separate project milestones are identified to communicate project status and accomplishments.   |  |  |  | | --- | --- | --- | | **Milestone Task** | **Start Date** | **End Date** | | Prototype Test Planning | Day 1 of month 1 |  | | Prototype Test Design |  |  | | Prototype Test Development |  |  | | Prototype Test Execution |  |  | | Prototype Test Evaluation |  | Last day of month 6 |     **6.****Deliverables**  The deliverables of the test activities as defined in this Test Plan are outlined in the table below.   |  |  |  | | --- | --- | --- | | **Deliverable** | **Review / Distribution** | **Due Date** | | Test Plan | Senior Project Mgmt Team |  | | Test Environment | - |  | | Test Suite | Internal Peer Review |  | | Test Data Sets | Internal Peer Review |  | | Test Scripts | Internal Peer Review |  | | Test Scripts | - |  | | Test Stubs, Drivers | - |  | | Test Defect Reports | Senior Project Mgmt Team |  | | Test Results | - |  | | Test Evaluation Report | Senior Project Mgmt Team | Last day of day 6 |     **6.1****Test Suite**  The Test Suite will define all the test cases and the test scripts which are associated with each test case.  **6.2****Test Logs**  It is planned to use RequisitePro to identify the test cases and to track the status of each test case. The test results will be summarized in RequisitePro as untested, passed, conditional pass, or failed. In summary, RequisitePro will be setup to support the following attributes for each test case:   * + Test status   + Build Number   + Tested By   + Date Tested   + Test Notes   It will be the responsibility of the System Tester to update the test status in RequisitePro.  Test results will be retained under Configuration Control.  **6.3****Defect Reports**  Rational ClearQuest will be used for logging and tracking individual defects. |

|  |
| --- |
|  |