# Test Design

# Owners and List of Contacts

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Email** | **Phone** | **Role** |
| Hemanth Kumar Mangala | [\*\*\*](mailto:mkovin@programmline.com) |  | Requirements analysis  Quality assurance |
| Prasoon Rana | [\*\*\*](mailto:kchermnykh@programmline.com) |  | Overall project direction  Communication with client  Technical consultant  Reporting |

## 1**. Introduction**

The primary goal of this document is to establish a plan for the activities that will verify **product** as a high quality product that meets the needs of business community. These activities will focus upon identifying the following:

* Items to be tested
* Testing approach
* Roles and responsibilities
* Release criteria
* Hardware
* Risk Assessment

## 2**. Items to Be Tested**

### 2.1 **Strategy of Testing**

Below is a list of services that testing may provide. Next to each service is the degree of testing that we will perform. Below are the valid level desired:  
Below is a list of services that testing may provide. Next to each service is the degree of testing that we will perform. Below are the valid level desired:  
Below is a list of services that testing may provide. Next to each service is the degree of testing that we will perform. Below are the valid level desired:

|  |  |
| --- | --- |
| **Level Desired** | **Service** |

|  |  |
| --- | --- |
| High | **End to End / Interface Testing:** End to End testing is testing all inputs (super-systems) and outputs (sub-systems) along with the application. A controlled set of transactions is used and the test data is published prior to the test along with the expected results. This testing ensures that the application will interact properly with the other systems. |

### 

### **2.2 Features to be Tested**

Below is a list of features that will be testedBelow is a list of services that testing may provide. Next to each service is the degree of testing that we will perform. Below are the valid level desired:  
Below is a list of services that testing may provide. Next to each service is the degree of testing that we will perform. Below are the valid level desired:  
Below is a list of services that testing may provide. Next to each service is the degree of testing that we will perform. Below are the valid level desired:  
Below is a list of services that testing may provide. Next to each service is the degree of testing that we will perform. Below are the valid level desired:  
Below is a list of services that testing may provide. Next to each service is the degree of testing that we will perform. Below are the valid level desired:  
Below is a list of services that testing may provide. Next to each service is the degree of testing that we will perform. Below are the valid level desired:  
Below is a list of services that testing may provide. Next to each service is the degree of testing that we will perform. Below are the valid level desired:  
Below is a list of services that testing may provide. Next to each service is the degree of testing that we will perform. Below are the valid level desired:  
Below is a list of services that testing may provide. Next to each service is the degree of testing that we will perform. Below are the valid level desired:

|  |  |  |  |
| --- | --- | --- | --- |
| **Business Requirements** | **Ref. No.** | **Requirements** | **Range** |

|  |  |  |  |
| --- | --- | --- | --- |
| Marketplace functionality, | RF.001 | Attendance Sheet  a. User will have option to create their attendance sheet and should  be able to navigate to it using ‘My Attendance’  b. User should be able to Punch IN and OUT their timing using option  ‘Punch IN/Punch Out’  c. In Morning user should be able to only Punch IN so can enter their  ‘in time’. Only when Punch IN is done Punch OUT options should  come.  d. With regularly Punching IN and Punching OUT, ‘Pay hours’ in the  ‘My Attendance sheet’ should reflect Total Pay hour.  i. Pay hours is a display field on ‘My Attendance’ sheet.  ii. It should be filled with total Pay hour the user.  iii. Example: 1 st Day: 4 hours, 2 nd Day: 3 hours  1. Pay hour display field should have 7 hours.  e. The same time should reflect when user clicks on day in  Attendance sheet with their Punch IN and Punch OUT Time.  i. First Day: 4 hours with punch in and Punch out time  ii. Second Day: 3 hours with punch in and punch out  time  f. Time should reflect in Users time zone. | High |

## **3. Testing Approach**

The system test team will begin designing their detailed test plans and test cases, as the development team is designing and coding. Manual testing will be done and test results will be published in issued folder.

The builds will be delivered to system test via Subversion drops coordinated by the development team. The development team will be responsible for installing the partial new builds into the existing structure of the system test environment, and updating the client machines if necessary. Build notes with all changes since the last drop and all files to be delivered will accompany each build drop.

Once the build is dropped by the development team, a series of scripts, called the Smoke Test, will be run to ensure that the shipment from development is in a state that is ready for testing. The Smoke Test scripts will test the basic functionality of the system. These scripts may be automated once they are successfully performed manually. If an excessive number of Smoke Test items fail, the product will be shipped back to development and no testing will begin until the Smoke Test passes.

Developers will perform all smoke tests on local version of system before publishing in repository. Developer is not able to publish any codes to repository if some of smoke test items fail.

Testers carry out testing according to strategy. Testers set severity for each issue and Project manager/Team Leader uses these to prioritize bugs fixing. Each day additional drops will be delivered to regression test. At the end of development there is a release for which system testing is carried out.

### 3.1 **Bug tracking system**

Test Lead will put new and reopened issues to <Project repository>/issues/new. Project manager and Team leader will assign new issues to project participants. Assigned tasks will be placed to <Project repository>/issues/assigned/<Person login>.

Developers will put fixed issues to <Project repository>/issues/fixed.

Test Lead will move resolved and closed issues to <Project repository>/issues/resolved.

## 4. Release Criteria

### 4.1 Test Case Pass/Fail Criteria

The feature will pass or fail depending upon the results of testing actions. If the actual output from an action is equal to the expected output specified by a test case, then the action passes. Should any action within a test case fail, the entire feature or sub-feature fails. The specific criteria for test case failure will be documented in test cases documents.

If a test case fails, it is not assumed that the code is defective. A failure can only be interpreted as a difference between expected results, which is derived from project documentation, and actual results. There is always the possibility that expected results can be in error because of misinterpretation, incomplete, or inaccurate project documentation.

Pass criteria:

1. All processes will execute with no unexpected errors
2. All processes will finish update/execution in an acceptable amount of time based on benchmarks provided by the business analysts and documented by the development team

Severity Codes are used to help Project manager/Team Leader to prioritize developers’ work in the test phase. They are assigned by testers. The following standard Severity Codes to be used for identifying defects are:

|  |  |  |
| --- | --- | --- |
| ***Severity Code Number*** | ***Severity Code Name*** | ***Description*** |
| 1 | Critical | Testing cannot proceed further within existing test cases (data corruption, crash, hang), there is not workaround. |
| 2 | High | The test case can be completed, but application incorrect output when valid information is input or test case completes with critical error but there is workaround |
| 3 | Medium | The test case can be completed and application correct output when valid information is input, but application incorrect output when invalid information is input.  (e.g. no special characters are allowed as part of specifications but when a special character is a part of the test and the system allows a user to continue, this is a medium severity) |
| 4 | Low | All test cases and passed as written, but there could be minor revisions, cosmetic changes, etc. These defects do not impact functional execution of system |

### 4.3 **Release to User Acceptance Test Criteria**

The release criteria necessary to allow the code to migrate to User Acceptance Testing are as follows:

* There are no open bugs with a severity 1 or 2
* Test cases scheduled for both Integration and system test phases have passed.
* Successfully passes the final regression testing.
* There are no discrepancies between the master setup and the version used during the final regression testing.

### 4.4 **Release to Production Criteria**

The release criterion necessary to allow the code to migrate to Production is as follows:

* There are no open bugs with a severity 1 or 2
* Test cases scheduled for both Integration and system test phases have passed.
* Successfully passes the final regression testing.
* There are no discrepancies between the master setup and the version used during the final regression testing.
* The User Acceptance Test was successfully completed
* The User Acceptance Criteria was met.

## **5. Software and Hardware**

### 5.1 Servers

|  |  |  |  |
| --- | --- | --- | --- |
| **Machine Name** | **Purpose** | **Owner** | **Hardware** |
| p-line | Test Server | Programmline | 2 x Intel Xeon 2,4-2,8 4Gb RAM 2 x 100Mbps(1000Mbps) network interface |

Below is a list of services that testing may provide. Next to each service is the degree of testing that we will perform. Below are the valid level desired:

### 5.2 **Server Configuration:**

P-line:

Platform

* MS Windows Server 2000

Database

* MS SQLServer 2000

Webserver

* Internet Information Server 5.5 or later

Interoperability

* SOAP 1.1

### **5.3 Clients:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Machine Name** | **Machine Location** | **Operating System** | **Hardware** |
| PC | \*\*\* | XP Professional SP2 English | P4-3.0-3.1 1Gb RAM |
| MAC | \*\*\* |  | P4-3.0-3.1 1Gb RAM |

## **6. Risks Assessment**

|  |  |
| --- | --- |
| **Risk ID** | **Risk description** |
| 1 | User interface will be developed with delay therefore tests - cases can be developed not qualitatively |
| 2 | Installation of test environment will demand a lot of time hence terms of testing can move |
| 3 | It will not be possible to receive a tool for performance testing hence the application can contain issues |