PinUp

Building motivation one task at a time

Test Plan

Group 2:

Xavier Davis, Nabeel Hussain, Fleury Keigni Di Satchou, Shawn Sokoloski

University of Maryland University College

CMSC 495

5/04/2018

Version # 1.3

Revision Sheet

Version	Date	Author	Description of Change
1.0	03/30/2018	Xavier Davis	Draft
1.1	03/30/2018	Fleury Keigni Di Satchou	Draft Reviewed
1.2	03/31/2018	Nabeel Hussain	Edits / Review
1.3	05/04/2018	Nabeel Hussain	Final Edits / Review

Approvers List: To track who has reviewed and signoff on the Test plan

Name	Role	Reviewer Initial	Reviewed Date
Fleury Keigni Di Satchou	Test Engineer	FK	05/04/2018
Xavier Davis	Project Manager Software Engineer	XD	05/04/2018
Nabeel Hussain	Project Manager Integration Engineer	NH	05/04/2018
Shawn Sokoloski	Technical Writer	SS	05/04/2018

Reference Documents: To clearly mark the document used as an input to create the test plan

Version	Date	Document Name
2.0	03/25/2018	High Level Requirements
2.0	03/30/2018	User’s Guide
3.0	05/04/2018	Updated User’s Guide

Table of Contents

1. INTRODUCTION

1.1. Purpose

This test plan describes the testing approach and overall framework that will drive the testing of the PinUp Version 1.0 or PinUp app. This document also introduces:

- Test Strategy: rules the test will be based on, including the givens of the project (e.g.: start / end dates, objectives, assumptions); description of the process to set up a valid test (e.g.: entry / exit criteria, creation of test cases, specific tasks to perform, scheduling, data strategy).

- Execution Strategy: describes how the test will be performed and process to identify and report defects, and to fix and implement fixes.

- Test Management: process to handle the logistics of the test and all the events that come up during execution (e.g.: communications, escalation procedures, risk and mitigation, team roster)

1.2. Project Overview

- PinUp Version 1.0 is a powerful web application tool that allows a platform for employers, parents, and sponsors to post small tasks that require payment outside of official payroll formats.

- With an internet enabled PC, users will have the ability to creates tasks and subaccounts once their login credential is verified by the server.

- The functionality of this module spans through the entire system of Stellar Lumens blockchain technology, making information secured and available anywhere, anytime.

- All information is subject to the user's policy, where he/she can only view the information he/she is authorized to.

- A user can only edit certain fields in his/her created subcontractor's module, deleting a task, maintaining the security of their subcontractor's financial information.

1.3. Audience

- Test Engineer - Test activities in the overall project schedule, reviews the document, tracks the performance of the test according to the task herein specified, approves the document and is accountable for the results. Ensures that the test plan and deliverables are in line with the design, provides the environment for testing and follows the procedures related to the fixes of defects.

- Project team - Provide input and recommendations on this document.

2. TEST STRATEGY

2.1. Test Objectives

- The objective of the test is to verify that the functionality of PinUp Version 1.0 works according to the specifications.

- The test will execute and verify the test scripts, identify, fix and retest all high and medium severity defects per the entrance criteria, prioritize lower severity defects for future fixing.

- The final product of the test is twofold:

    - A production-ready software;

    - A set of stable test scripts that can be reused for Functional and User's test execution.

2.2. Test Scope

Key scope:

The application must be tested to meet the scope and out-of-scope listed in the application's High-Level Requirements document. The AWS application test analysis tool is still pending approval by the project team, because most of the tools require fees, meanwhile this is for academic course work

General:

- Exploratory Testing would be carried out once the build is ready for testing
- Performance testing is not considered for this estimation.
- All the defects would come along with a snapshot JPEG format
- The test team assumes all necessary inputs required during test design, and the execution will be supported by other team members appropriately.
- Test case design activities will be performed by the Test Engineer
- The dev team will provide plans to fix defects based on the defect/bug meetings during each phase cycle.

- Any defect fixes planned will be shared with the Project Managers
- Project Managers will review and sign-off on all test deliverables.
- The project will provide test planning, test design and test execution support for the system, the project and the testing processes.

2.3. Test Principles

- Testing will be focused on meeting the business objectives, cost efficiency, and quality.
- There will be common, consistent procedures for all teams supporting testing activities.
- Testing processes will be well defined, yet flexible, with the ability to change as needed.
- Testing activities will build upon previous stages to avoid redundancy or duplication of effort.
- Testing environment and data will emulate a production environment as much as possible.
- Testing will be a repeatable, quantifiable, and measurable activity.
- Testing will be divided into distinct phases, each with clearly defined objectives and goals.
- There will be entrance and exit criteria.

2.4. Data Approach

In functional testing, PinUp 1.0 will contain pre-loaded test data and which is used for testing activities.

2.5. Scope and Levels of Testing

2.5.1. Exploratory

Purpose - the purpose of this test is to make sure critical defects are removed before thenext levels of testing can start.

Scope - First level navigation, dealer, and admin modules

Method - this exploratory testing is carried out in the application without any test scripts

and documentation

Timing - at the beginning of each cycle.

2.5.2. Functional Test

Purpose - Functional testing will be performed to check the functions of application. The functional testing is carried out by feeding the input and validates the output from the application.

Scope - User must have access to an established internet connection with accessibility to the world wide web domain. Preload data will be available in the system (MongoDB Atlas) for testing.

Method - The test will be performed according to Functional scripts

Timing - After Exploratory test is completed.

2.5.3. User Acceptance Test (UAT)

Purpose - This test focuses on validating the business logic. It allows the end users to complete one final review of the system prior to deployment.

Testers - the UAT is performed by the end users (sponsors/subcontractors).

Method - Since the users (sponsors/ subcontractors) are the most indicated to provide input around business needs and how the system adapts to them, it may happen that the users do some validation not contained in the scripts. UAT test cases based on the inputs from End user cannot be written at this time.

Timing - After all other levels of testing (Exploratory and Functional) are done. Only after this test is completed the product can be released to production.

3. TEST EXECUTION & DEFECTS HANDLING

3.1. Entry and Exit Criteria

- The entry criteria refer to the desirable conditions in order to start test execution; only the migration of the code and fixes need to be assessed at the end of each cycle.

- The exit criteria are the desirable conditions that need to be met in order proceed with the implementation.

- Entry and exit criteria are flexible benchmarks. If they are not met, the test team will assess the risk, identify mitigation actions and provide a recommendation. All this is input to the project manager for a final “go-no go” decision.

- Entry criteria to start the execution phase of the test: the activities listed in this Test Plan of the schedule are 100% completed.
- Test Engineer must ensure the following exit criteria:
    - 100% Test Scripts executed.
    - 95% pass rate of Test Scripts.
    - No open Critical and High severity defects.
    - 95% of Medium severity defects have been closed.
    - All remaining defects are either cancelled or documented as Change Requests for a future release.
    - All expected and actual results are captured and documented with the test script.
    - All test metrics collected based on reports from a defined AWS application test analysis tool, approved by the team.

3.2. Test Cycles

There will be two cycles for functional testing. Each cycle will execute all the scripts.
The objective of the first cycle is to identify any blocking, critical defects, and most of the high defects. It is expected to use some work-around in order to get to all the scripts.
The objective of the second cycle is to identify remaining high and medium defects, remove the work-around from the first cycle, correct gaps in the scripts and obtain performance results.
UAT test will consist of one cycle.

3.3. Validation and Defect Management

It is expected that the testers execute all the scripts in each of the cycles described above. However, it is recognized that the testers could also do additional testing if they identify a possible gap in the scripts. This is especially relevant in the second cycle, when the test engineer joins the project manager in the execution of the test, since the project manager have a deeper knowledge of the business processes. If a gap is identified, the scripts and traceability matrix will be updated and then a defect logged against the scripts.
The defects will be tracked through AWS test analysis tool only. The Project team will gather information on a daily basis from AWS test analysis tool and request additional details from the test engineer. The team will work on fixes.

It is the responsibility of the test engineer to open the defects, link them to the corresponding script, assign an initial severity and status, retest and close the defect; it is the responsibility of the project team to review the severity of the defects and facilitate with the technical team the fix and its implementation, communicate with testers when the test can continue or should be halt, request the tester to retest, and modify status as the defect progresses through the cycle; it is the responsibility of the technical team to review AWS test analysis tool on a daily basis, ask for details if necessary, fix the defect, communicate to the test engineer the fix is done, implement the solution per the test engineer request.

Defects found during the Testing will be categorized according to the bug-reporting tool and the categories must be in the form:

Severity	Impact
1 (Critical)	▪ This bug is critical enough to crash the system, cause file corruption, or cause potential data loss. ▪ It causes an abnormal return to the operating system (crash or a system failure message appears). ▪ It causes the application to hang and requires re-booting the system.
2 (High)	▪ It causes a lack of vital program functionality with workaround.
3 (Medium)	▪ This Bug will degrade the quality of the System. However, there is an intelligent workaround for achieving the desired functionality for example through another screen. ▪ This bug prevents other areas of the product from being tested. ▪ However other areas can be independently tested.
4 (Low)	▪ There is an insufficient or unclear error message, which has minimum impact on product use.
5 (Cosmetics)	▪ There is an insufficient or unclear error message that has no impact on product use.

3.4. Test Metrics

Test metrics to measure the progress and level of success of the test will be developed and shared with the project manager for approval.

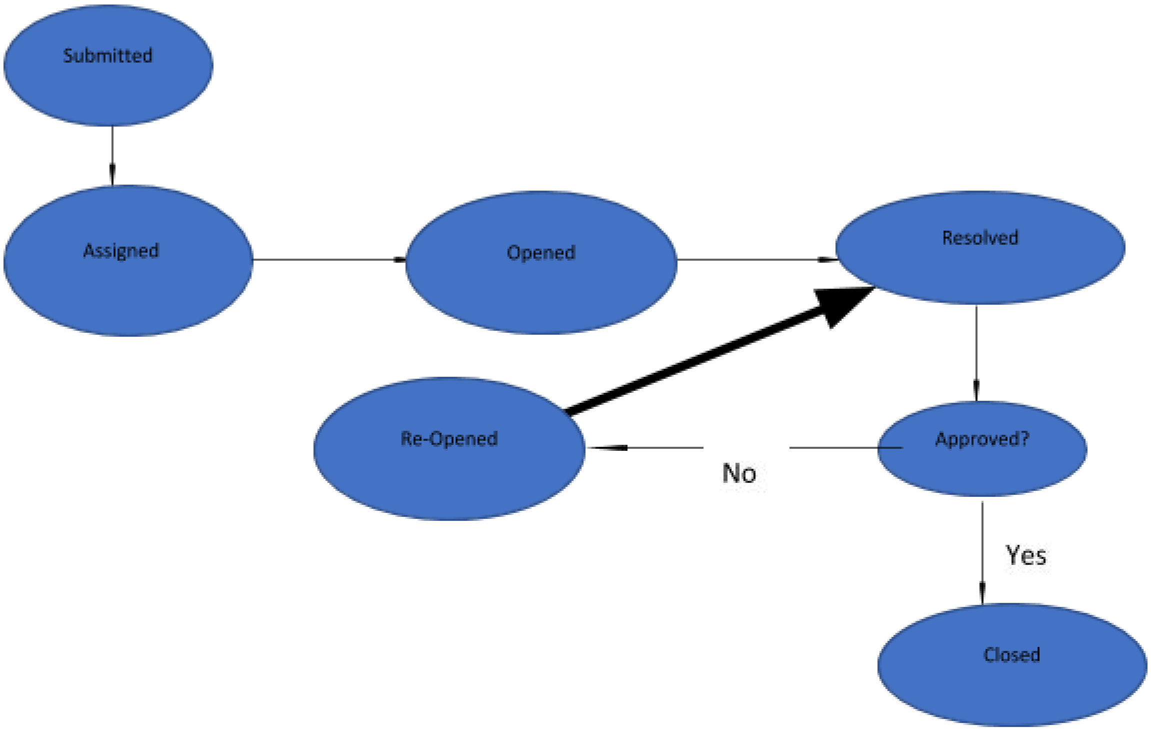
3.5. Defect Tracking & Reporting

A Defect will pass through the following stages:

Status	Description
Submitted	Defect created by Tester and Assigned
Opened	Defect is determined as a valid defect

Assigned	Assigned to a developer to fix
Resolved	Developer has fixed defect
Approved?	Has the defect been properly fixed?
Closed	No further action required – Closed by Tester
Reopened	When the defect is re-tested but not ok

Following flowchart depicts Defect Tracking Process:



3.6. Defect Review Meetings

During Test execution, regular (usually daily) meetings will be held between the Project Manager, Lead Tester, and the developers. Other parties, e.g. Integration Engineers or Technical Writers may also be consulted on specific issues.

The main objectives of the meeting will be to:

- Review all new application and environment defects raised in order to agree that they are correctly categorised and prioritised.

- Monitor current status of all un-resolved defects
- Discuss defects which are obstructing tests
- Discuss requirements for any unscheduled code deployments or data refreshes to the test environment from live

In the event of any disagreement between the parties regarding the categorisation of defects, the matter shall be resolved by the project managers.

4. TEST MANAGEMENT PROCESS

4.1. Test Management Tool

- Any approved AWS software analysis testing tool. All testing artifacts such as Test cases, test results are updated in the approved AWS management tool.
- Project specific folder structure will be created in AWS management tool to manage the status of this project.
- Each resource in used by the team will be provided with Read/Write access to add/modify Test cases in AWS management tool.
- All test cases are written directly into AWS management tool. Any change to the test case will be directly updated in the AWS management tool.
- Various reports can be generated from AWS management tool to provide status of Test execution. For example, Status report of Test cases executed, Passed, Failed, No. of open defects, Severity wise defects etc.

4.2. Test Design Process

- The test engineer will understand each requirement and prepare corresponding test case to ensure all requirements are covered.
- Each Test case will be mapped to Use cases to Requirements as part of Traceability matrix.
- Each of the Test cases will undergo review by the project team and the review defects are captured and shared. The test engineer will rework on the review defects and finally obtain approval and sign-off from the project manager.

4.3. Test Execution Process

Automated execution- Automated execution process will be meets will reference the project team approved AWS management tool.

Manual Execution-

GUI Testing:

Test Objective:	Ensures requirements such as web page appearance, design structure, required fields displays the appropriate messages, and ease of use.
Technique:	Manual testing through observation, input of incorrect login credentials and flow of event
Completion Criteria:	Web page must be capitative for users. Appropriate error messages must be displayed. Flow of event must be met per High Level Requirement document.
Special Considerations:	Out of scope requirements listed on High Level document will not be tested.

Functional Testing:

Test Objective:	To verify that PinUp application performs and functions correctly according to design specifications.
Technique:	Check the core application functions such as registering users, link subaccounts, login, task creation and payout process.
Completion Criteria:	PinUp must meet all functional requirement as defined in User's Guide and High Level Required document.
Special Considerations:	An actor must have access to an established internet connection with accessibility to the world wide web domain. Preload data will be available in the system (MongoDB Atlas) for testing. Using a laptop or desktop hardware, system interactions are possible.

Backend Testing:

Test Objective:	Ensures system backend is secured.
Technique:	Will inject NoSQL text into MongoDB atlas to ensure consistency restriction into the backend.

Completion Criteria:	Communication with the backend will be secured and consistent.
Special Considerations:	Only signup login fields will be used.

4.4. Test Risks and Mitigation Factors

Risk	Prob.	Impact	Mitigation Plan
SCHEDULE Testing schedule is tight. If the start of the testing is delayed due to design tasks, the test cannot be extended beyond the UAT scheduled start date.	High	High	The testing team can control the preparation tasks (in advance). Some buffer has been added to the schedule for contingencies, although not as much as best practices advise.
RESOURCES Not enough resources, resources on boarding too late (process takes around 15 days.	Medium	High	Holidays and vacation have been estimated and built into the schedule; deviations from the estimation could derive in delays in the testing.
DEFECTS Defects are found at a late stage of the cycle or at a late cycle; defects discovered late are most likely be due to unclear specifications and are time consuming to resolve.	Medium	High	Defect management plan is in place to ensure prompt communication and fixing of issues.
Delayed Testing Due To new Issues	Medium	High	During testing, there is a good chance that some "new" defects may be identified and may become an issue that will take time to resolve. There are defects that can be raised during testing because of unclear document specification. These defects can yield to an issue that will need time to be resolved.

			If these issues become showstoppers, it will greatly impact on the overall project schedule. If new defects are discovered, the defect management and issue management procedures are in place to immediately provide a resolution.

5. TEST ENVIRONMENT

Primary:

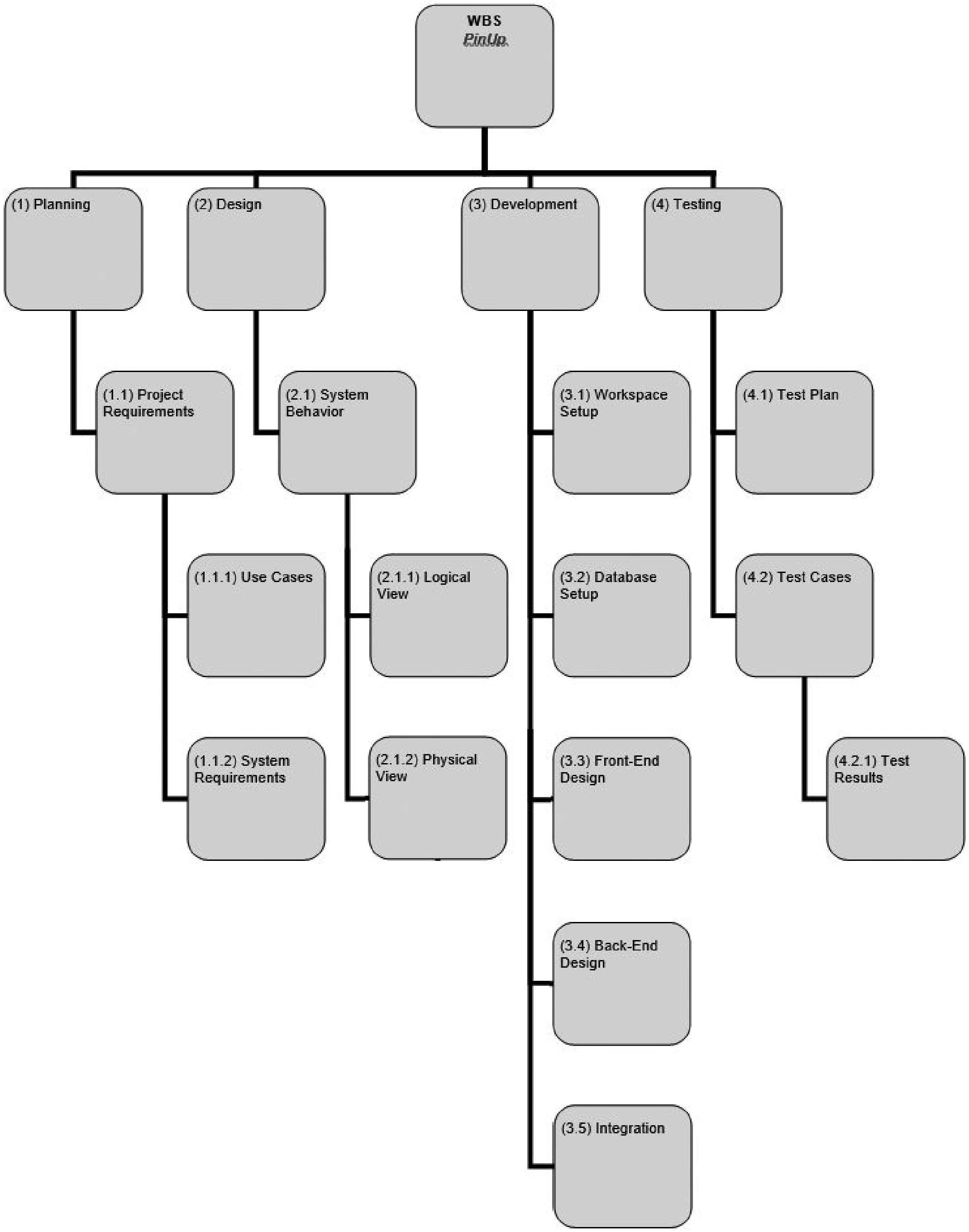
Programming Languages	**Dependencies**	**Cloud Hosting Service**
HTML, CSS, JavaScript	Node.js, jQuery, Bootstraps 4, npm, Mongoose,	AWS, MongoDB Atlas

Other:

- A windows environment with Internet Explorer 8, 9 and 10, and with Firefox 27.0, as well as Google.
- Chrome 65.0 and later should be available to the Test Engineer.

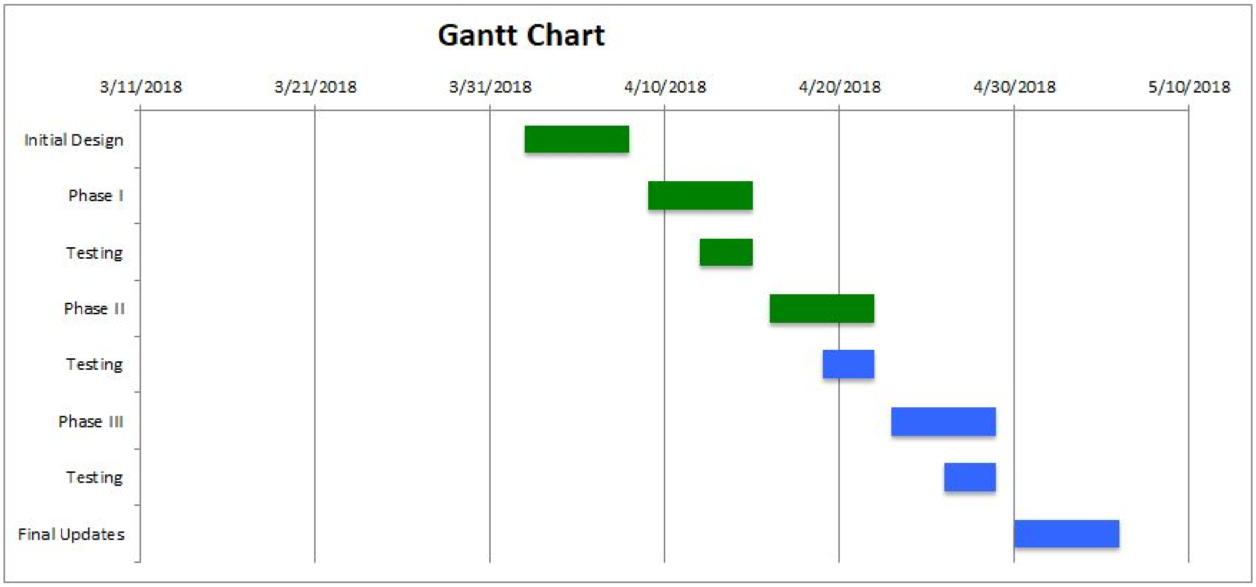
6. TEST SCHEDULE

6.1. Work Breakdown Structure (WBS)



6.2. Gantt Chart

Task Name	Start	End	Duration (days)
Initial Design	4/2/2018	4/8/2018	6
Phase I	4/9/2018	4/15/2018	6
Testing	4/12/2018	4/15/2018	3
Phase II	4/16/2018	4/22/2018	6
Testing	4/19/2018	4/22/2018	3
Phase III	4/23/2018	4/29/2018	6
Testing	4/26/2018	4/29/2018	3
Final Updates	4/30/2018	5/6/2018	6



APPENDIX A: KEY CODES

A.1. Priority Level Table

Priority	Description
1	Prevents any meaningful testing being carried out
2	Stops a significant area of testing from progressing to completion
3	Stops a specific test being completed
4	Defect which does not stop a test being run to completion
5	Query resulting from an unexpected occurrence that does not impact on test

A.2. Severity Level Table

Severity	Description
1	**Critical** – System cannot go live
2	**High** – The defect is significant with part of the system not working as specified
3	**Moderate** – the system does not meet stated requirements and impacts on the use of a significant part of the system
4	**Low** – does not meet sta
5	**Minor** - does not meet stated requirements but has no adverse impact on the use of the system

A.3. Resolution Codes

Every defect must be closed down in the defect tracking system with an associated Resolution Code from the list of valid values:

- Change Request/Next Release
- Code Error
- Data Error
- Environment Build Problem
- Error already in production
- Raised in Error
- Tolerate