Test Plan Document

For

# e-Care Clinical System

Version 1.1

Prepared by

Drishti Shishodiya

Akshay Shinde

Guozhi Tang

YuSen Wang

**TABLE OF CONTENTS (TOC)**

1.0 Introduction

2.0 Objectives and Tasks  
2.1 Objectives  
2.2 Tasks

3.0 Scope

4.0 Entry and Exit Criteria  
4.1 Entry criteria  
4.2 Exit criteria

5.0 Testing Strategy  
5.1 Unit Testing  
5.2 Integration Testing  
5.3 System Testing  
5.4 User Acceptance Testing  
5.5 Performance Testing  
5.6 Regression Testing

6.0 Test Schedule

7.0 Control Procedures

8.0 Features to Be Tested

9.0 Features Not to Be Tested

10.0 Resources/Roles & Responsibilities

11.0 Schedules

12.0 Dependencies

13.0 Risks/Assumptions

14.0 Tools

15.0 Approvals

1. **INTRODUCTION**

The e-Care Clinicals application is meant to replace the legacy pro-care application. It has comprehensive functionalities to help nurses perform their tasks more efficiently and easily. It performs functions like patient admission, patient treatment, reporting results, sending and receiving data with spectra labs, managing document imaging and demographic information of patients and dealing with Medicare/Medicaid and commercial insurances for financial claims and charges.

**2.0 OBJECTIVES AND TASKS**

**2.1 Objectives**

This document describes the plan for testing the e-care Clinicals System application and supports the following objectives:

* Identifying existing project information and the software to be tested.
* Recommend and describe the testing strategy to the team.
* Identify the required resources and provide an estimate of the test efforts.
* Detailing the features, responsibilities, dependencies and risks involved.

**2.2 Tasks**

Below are the test related tasks for testing the e-Care Clinicals system:

|  |
| --- |
| **Plan Test** |
| Identify Requirements to Test |
| Assess Risk |
| Develop Test Strategy |
| Identify Test Resources |
| Create Schedule |
| Generate Test Plan |
| **Design Test** |
| Analyze functional requirements |
| Develop Test Suite  Create Test Scenarios |
| Identify and Describe Test Cases |
| Identify and Structure Test Scripts |
| Review and Access Test Coverage |
| **Implement Test** |
| Setup Test Environment |
| Record or Program Test Scripts |
| Develop Test Stubs and Drivers |
| Identify Test-Specific functionality |
| Create Test data |
| **Execute Test** |
| Execute Test Scripts |
| Evaluate Execution of Test |
| Recover from Halted Test |
| Verify the results |
| Investigate Unexpected Results |
| Log Defects |
| **Evaluate Test** |
| Evaluate Test-Case Coverage |
| Evaluate Code Coverage |
| Analyze Defects |
| Determine if Test Completion Criteria and Success Criteria have been achieved |
| Create Test Evaluation Report |

1. **SCOPE**

The overall purpose of testing the e-Care clinicals system is to ensure that the application meets all the technical, functional and business requirements. The approach described in this document provides the framework for all testing related to this application. The individual modules and interface between the following modules will be tested:

* Patient admission
* Patient treatment
* Reporting
* Financial charges
* Financial claims
* Spectra labs
* Document imaging

1. **ENTRY AND EXIT CRITERIA**

**4.1 Entry Criteria**

Entry criteria can be defined as the specific conditions that should be present before a process of testing can begin. They include:

* Requirements should be defined, approved and clear.
* Code should be ready and testable
* Appropriate test environment should be set.
* Test documents should be ready or in progress.
* Test tools should be set.
* Resources should be available to test.

**4.2 Exit Criteria**

Exit criteria can be defined as the specific conditions that should be fulfilled prior to completing the software testing life cycle. It is used to determine whether a given test activity has been completed or not. The conditions include:

* All critical test cases should be passed.
* Complete functional coverage should be achieved.
* Major functional/business flows should be executed successfully by leveraging various test inputs.
* All the high priority defects should be identified and fixed.
* All the show stopper or blocker defects should be fixed and none of the identified critical or severity 1 defects should be in open status.
* All the high priority defects should be retested and closed to execute the corresponding Regression scenarios successfully.

The following outputs are achieved through the exit criteria:

* Test log
* Test incident report
* Test summary report

**5.0 TESTING STRATEGY**

The testing phase for e-Care clinical system will begin with creating test documents like test plan, test scenarios, test cases and test data. After all the documents are created, test environment and tools are ready, and the code is delivered to test, the application will go through different types of testing. Each individual module of the application will go through unit testing. Then combining multiple units together, integration testing will be performed and then system testing will be performed which ensures a compatibility check of the application with the entire system.

HP-ALM QC will be used as a test management tool to collectively store the details of the requirements, test suite, test cases, test execution and defects.

Describe the overall approach to testing. For each major group of features or feature combinations, specify the approach which will ensure that these feature groups are adequately tested. Specify the major activities, techniques, and tools which are used to test the designated groups of features.

**5.1 Unit Testing**

**Definition:** Unit testing is a level of software testing where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output. It is the first level of testing and is performed prior to the integration testing. For the e care clinical system modules which are considered as a single unit are: patient admission, patient treatment, financial claims, reporting, financial charges, insurance, spectra labs and document imaging.

**Participants:** Unit testing will be performed by the developers in the team.

**Methodology:** Unit testing will be performed by using white box testing technique. In this the developer will look into the code of individual modules mentioned above. Separate test scripts will be created and executed to ensure that they pass individually. After receiving successful unit testing results the developer will deploy the code into testing.

**5.2 Integration Testing**

**Definition: Integration testing** is a level of software testing where individual units are combined and tested as a group. The purpose of this is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing. For the e care clinical system, the integration testing is a very important aspect as there are multiple modules interacting with each other. So, this will test whether the clinical system and financial system modules are both working as expected when interacting with each other. And similarly, will check the compatibility between other modules.

**Participants:** Integration testing will be performed by the testers in the team.

**Methodology:** Integration testing will be performed by using black box testing technique. In this the tester will look into the interfaces between the modules. Create test cases and execute to verify whether the data is easily flowing from one module to another.

**5.3 System Testing**

**Definition: System testing** is a level of software testing where a complete and integrated software is tested. The purpose of this test is to evaluate the system’s compliance with the specified requirements. Also, the functionalities of the system are tested from an end-to-end perspective. It includes both functional and Non-Functional testing. For the e care clinical system, the system testing will include verification of the whole application with the general system requirements.

**Participants:** System testing is performed by the testers in the team.

**Methodology:** System testing will be performed by using black box testing technique. In this the tester will look into the interaction of the application with the various system functional and non-functional properties.

**5.4 User Acceptance Testing**

**Definition:** User acceptance testing (UAT) is the last phase of the software testing process. During UAT, actual software users test the software to make sure it can handle required tasks in real-world scenarios, according to specifications. UAT is one of the final and critical software project procedures that must occur before newly developed software is rolled out to the market. It is also known as Beta testing. For the e care clinical system, the UAT testing by the nurses will help in understanding the effectiveness or issues in the application.

**Participants:** UAT testing is performed by the testers in the team.

**Methodology:** UAT testing will be performed by some business users at their own environment. The users will perform each functionality which is required in their daily tasks and check whether the application is meeting their expectations. They try multiple positive and negative scenario combinations and test the application thoroughly.

**5.5 Performance Testing**

**Definition:** Performance testing is a non-functional testing technique performed to determine the system parameters in terms of responsiveness and stability under various workload. Performance testing measures the quality attributes of the system, such as scalability, reliability and resource usage. For the e care clinical system, the performance needs to be tested as it is very important that the application has lesser response time and is able to handle the load of multiple users at a time.

**Participants:** Performance testing is performed by the testers in the team.

**Methodology:** Performance testing will be performed by using certain tools to insure the quality of the system. Load and stress testing are some of the important techniques used in this. Also, various attributes of performance testing are: speed, scalability, stability and reliability.

**5.6 Regression Testing**

**Definition:** Regression Testing is defined as a type of software testing to confirm that a recent program or code change has not adversely affected the existing features.It is nothing but full or partial selection of already executed test cases which are re-executed to ensure existing functionalities work fine. It ensures that old code still works once the new code changes are done. For the e care clinical system regression testing plays an important role as there are many different modules interacting with each other.

**Participants:** Regressiontesting is performed by the testers in the team.

**Methodology:** Regression testing is required when there is a change in requirement and the code is modified as per the requirement, any new feature is added to the software, defect fixing and performance issue fix. All the critical test cases will be selected and re-executed to ensure that the other functionalities are also working properly along with it.

**6.0 TEST SCHEDULE**

Testing of the e-Care Clinical Application 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** | **Effort (pd)** | **Start Date** | **End Date** |
| Prototype Test Planning | 2 | December 12 | December 15 |
| Prototype Test Design | 3 | December 15 | December 18 |
| Prototype Test Development | 4 | December 19 | December 23 |
| Prototype Test Execution | 3 | December 24 | December 26 |
| Prototype Test Evaluation | 1 | December 29 | December 29 |

**7.0 CONTROL PROCEDURES**

**Problem Reporting:**

Incident log entries will be made for all bugs found during testing. The log will be used to track the status of the bugs.

Defect Tracking Process

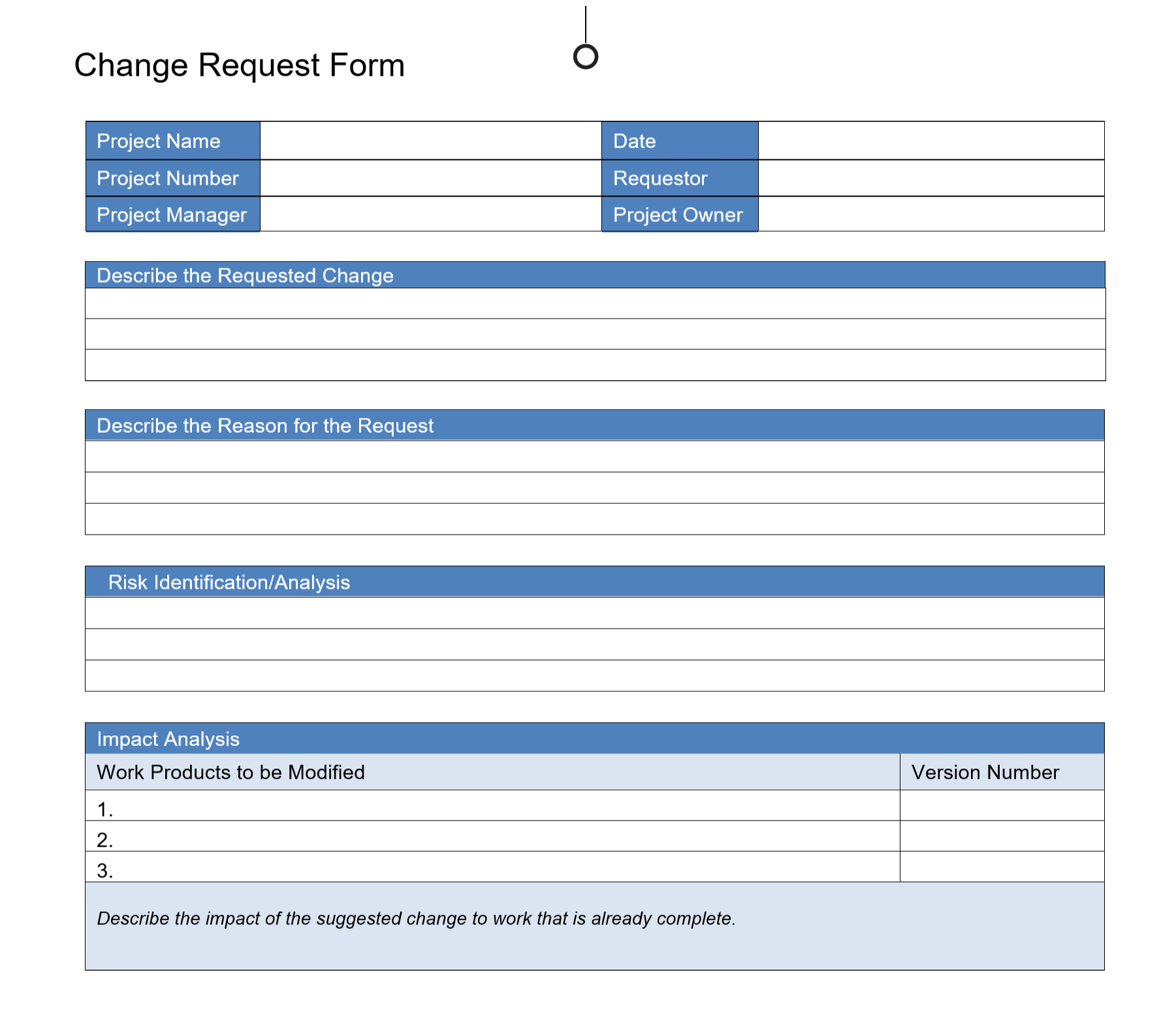
|  |  |
| --- | --- |
| **Summary:** | Screen name and short description about the defect being reported, usually providing key words with which to identify and/or search for the defect. |
| **Detected By:** | Auto populates with the User ID of person logged in. |
| **Detected on Date:** | Auto populates with current date. |
| **Severity:** | Describes the degree of impact that a defect has on the operation of the application. |
| **Assigned To:** | Individual being assigned the defect for fixing. |
| **Detected in Build:** | Build ID in which the defect was found. Build ID is an identifier for the code release, assigned by Web Development. |
| **Fixed in Build:** | Build ID in which the defect is fixed. Build ID is an identifier for the code release, assigned by Web Development. |
| **Priority:** | This field describes the impact the defect has on the work in progress and the importance and order in which a bug should be fixed. |
| **Status:** | Indicates the existing state of a defect, auto populates with “New” |
| **Description:** | Enter description of defect. Add individual steps to reproduce. Include all steps and screens that were accessed. Enter exact words of the error message. |
| **Email Defect:** | After entering defect, right-click on it and select email to send to assigned developer. |
| **Defect resolution process:** | When the defect is opened, it is assigned to the appropriate person, status is changed to “Assigned”.  Once the defect is fixed:   1. The developer to whom the defect is assigned will update the defect comments to document the fix that was made. User ID and Date is automatically added to the defect by clicking on “Add Comment”. 2. The developer to whom the defect is assigned will change the status to “Fixed”, and will change the “Assigned To” field to the tester or defect manager. 3. The tester will retest the submitted defect. 4. If defect passes the retest, the tester or defect manager will change Status to “Closed”. 5. If the defect is not fixed, the tester will change the Status to “Assigned” and enter the UserID of the developer in the Assigned To field. 6. Once the defect has been verified as fixed, the project manager (or defect manager) will update the status to “Closed”. |

|  |  |
| --- | --- |
| **DEFINITIONS FOR DEFECT PRIORITY AND SEVERITY** | |
| **PRIORITY:** This field describes the impact the defect has on the work in progress and the importance and order in which a bug should be fixed. This field is utilized by the developers and test engineers to prioritize work effort on the defect resolution. | |
| **1 – Urgent Blocks Work** | Further development and/or testing cannot occur until the defect has been resolved. |
| **2 – Resolve ASAP** | The defect must be resolved as soon as possible because it is impairing development and/or testing activities. |
| **3 – Normal Queue** | The defect should be resolved in the normal prioritization and completion of defect resolution. |
| **4 – Low Priority** | The defect is an annoyance and should be resolved, but it can wait until after more serious defects have been fixed. |
| **5 – Trivial** | The defect has little or no impact to development and/or testing work. |
|  | |
| **SEVERITY:** This field describes the degree of impact that a defect has on the operation of the application. | |
| **1 – Critical** | Critical loss of function. The defect results in system crashes, the failure of a key subsystem or module, a corruption or loss of data, or a severe memory leak. |
| **2 – Major** | Major loss of function. The defect results in a failure of the system, subsystem, or module, but the defect does not result in the corruption or loss of significant data. |
| **3 – Moderate** | Moderate loss of function. The defect does not result in a failure of the system, subsystem, or module, but the defect may cause the system to display data incorrectly, incompletely, or inconsistently. |
| **4 – Minor** | Minor loss of function, or another problem where a workaround is present. There are no data integrity issues. |
| **5 – Usability** | The defect is related to the system usability, is the result of non-conformance to a standard, or is related to the aesthetics of the system. There is no loss of system function. |
| **6 – Enhancement** | The defect is a request for an enhancement, i.e. it is not within the scope of the current project effort. |

**Change Requests:**

Any modifications to the requirements must be done through change requests, which will ensure the proposed change is fully reviewed before being incorporated into “e-Care Clinical Application”.

The Change Request template should be as below



**8.0 FEATURES TO BE TESTED**

“e-Care Clinical System” will be tested for its functionality. Testing include – (1) Unit testing, (2) Integration testing, (3) System testing and (4) User Acceptance Testing (UAT). Each test is scheduled to take one week with the Customer Acceptance test on 11-11-2008.

The testing includes testing for several functions like

* Login feature
* Admission of Patient
* Sending Patient Information to Spectra Labs
* Sending Lab Results back to e-Care Clinical System
* Sending claims to the Medicaid/Commercial Insurances
* Receiving payments from the Medicaid/Commercial Insurances
* Sending Treatment/Lab Charges to the Document Imaging Center.
* Generating Daily Reports

**9.0 FEATURES NOT TO BE TESTED**

Performance of the project “e-Care Clinical System” for a cluster of large number of patients is out of scope of this project at this point of time.   
**10.0 RESOURCES/ROLES & RESPONSIBILITIES**

Table below describes the Roles, Responsibilities and Resource Name for the testing of the Project “e-Care Clinical”

|  |  |  |
| --- | --- | --- |
| **Role** | **Responsibilities** | **Resource Name(s)** |
| Testers | * Plan testing activities * Execute Test Cases * Find, report and track defects * Measure test effort * Analyze results | John and Josh |
| **Developers** | * Deliver complete builds of the application * Provide Testers with feedback regarding changes, new functionality * Provide expertise and knowledge of the application-under-test * Eliminate agreed upon defects | Aaron |
| **Business Analysts** | * Interview Users * Create Business Requirements * Create Test Scenarios, Test Cases | Amol and Rahul |
| **Users** | * Describe and review Business Requirements * Describe and review user profiles * Perform Customer Acceptance Testing (CAT) | Carla, Cindy, Maggie and Martina |
| **Desktop Administrators** | * Installation of software * Troubleshooting of hardware/software * Information regarding standard desktop | Amit |

**11.0 SCHEDULES**

**Major Deliverables:**

|  |  |  |
| --- | --- | --- |
| **Deliverable** | **For** | **Date / Milestone** |
| Test Plan | Project Manager; QA Director; Test Team | 12/06/2018 |
| Traceability Matrix | Project Manager; QA Director | 12/05/2018 |
| Test Results | Project Manager | 12/06/2018 |
| Test Status report | QA Manager, QA Director | 12/06/2018 |
| Metrics | All team members | 12/06/2018 |

**12.0 DEPENDENCIES**

## **Personnel Dependencies**

The test team requires experience testers to develop, perform and validate tests. The test team will also need the following resources available: Application developers and Payroll Clerks.

## **Software Dependencies**

The source code must be unit tested and provided within the scheduled time outlined in the Project Schedule.

## **Hardware Dependencies**

The Mainframe, 10 PCs (with specified hardware/software) as well as the LAN environment need to be available during normal working hours. Any downtime will affect the test schedule.

## **Test Data & Database**

Test data (mock employee information) & database should also be made available to the testers for use during testing.

**13.0 RISKS/ASSUMPTIONS**

The following risks haven been identified and the appropriate action identified to mitigate their impact on the project.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Risk | Impact | Trigger | Mitigation Plan |
| 1 | Changes to the users’ requirements | High | Late delivery date | Periodic meeting will be conducted to communicate with project sponsors and stakeholders regarding product’s requirements. |
| 2 | Stakeholders not satisfied with functionality may negate the features already developed and we may redesign the application | High | Redesign application | Consistently report project’s status and features developed to the stakeholders. |
| 3 | Late delivery due to poor performance of the development teams | Medium | Product did not get delivered on schedule | Check with the project manager regularly and make sure he or she keeps on track of the development status. |

14.0 TOOLS  
  
We will be using ‘JIRA’ as our bug tracking tool. Jira is a [commercial software](https://en.wikipedia.org/wiki/Commercial_software) product that can be [licensed](https://en.wikipedia.org/wiki/Software_licensing) for running on-premises or available as a hosted application

Automation tools to be used during testing are as below:

- Selenium with Cucumber framework  
- REST Assured API

**15.0 APPROVALS**

By signing below, I acknowledge that I have read the entire contents of this document and accept the document in this form as reasonably fulfilling the goals described in the section titled Document Purpose. I further agree that this will constitute the document of record and cannot be changed without review and acknowledgement of the groups shown below:

|  |  |  |  |
| --- | --- | --- | --- |
| **Group / Role** | **Approver Name** | **Approver Signature** | **Date Approved** |
| QA Manager | Drishti Shishodiya | Drishti S. | 12/03/2019 |
| QA Lead | Wilson | Wilson | 12/01/2019 |
| QA Lead | Karson | Karson | 12/01/2019 |
| Automation Architect | Akshay Shinde | Akshay S. | 11/29/2019 |