Test Plan

**ChequeMeOut**

notsirkApps

**Prepared By**

|  |  |
| --- | --- |
| DOCUMENT OWNER(S) | Organization role |
| Kriston Sanders | Project Manager |

**Version Control**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author(s) | Change description |
| 1.0 | 5/28/20 | Kriston Sanders | Document created |

**Document Reference**

|  |  |
| --- | --- |
| DOCUMENT Name | File location |
|  |  |

**Definitions and Acronyms**

|  |  |
| --- | --- |
| Acronym | Definition |
|  |  |

**Approvals**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Approving party | Signature |
| 1.0 |  | Kriston Sanders | Kriston Sanders |

**Test-plan identifier**

ChequeMeOut\_0001

| 1. Introduction   Introduction or summary includes the purpose and scope of the project  Example: The objective of this document is to test the functionality of the ‘ProjectName’ |
| --- |
| * 1. **Background**   …   * 1. **Testing Objectives**   … |

| 1. Test items   A list of test items which will be tested  Example: Testing should be done on both front end and back end of the application on the Windows/Linux environments. |
| --- |
| * 1. **In Scope**  1. **Functional Scope**   Table?   1. **Non-functional Scope**   Table?   * 1. **Out of Scope**   [It is important to clearly define (at a high level) all of the testable components of the solution that will not be tested. These should include infrastructure, functional subsets, non-functional requirements, and software modules. Specific testing activities (such as load and performance testing, penetration tests, etc) should also be listed. Brief reasoning behind why the items have been de-scoped should be included].  Table?   * 1. **Testing Exclusions**   [A test exclusion is an element of the SUT that has not been de-scoped but which will not be tested by this plan due to (usually) a logistical issue. These may include activities such as report/letters distributed by fax/email/sms and may reference the test phase/level that will be responsible for conducting this testing]. |

| 1. Approach   The overall strategy of how testing will be performed. It contains details such as Methodology, Test types, Test techniques etc.,  Example: We follow Agile Methodology in this project  All testing conducted by, or for, the KT Program complies with the Master Test Strategy. This DTP has been created to define the test activities documented in the Master Test Plan  [Insert Release Name and Test Activity] will be tested using the following approach:  [List (in detail) the key elements that make up the strategy you will use to deliver the required test objectives. Wherever possible link specific activities to the relevant test objective. Change/edit the accompanying descriptions to suit the individual release]. |
| --- |
| * 1. **Naming Conventions for Function Groups and Functions**   [Use the Configuration Management naming conventions to define the structure used for the test project nomenclature. If HP Quality Center is to be used ensure the nomenclature rules also align with the requirements defined for it. The Test case ID should be agreed with the naming convention.  Rename this section as required:  … for Function Groups and Functions  … for Scenarios and Usecases  etc]   * 1. **Test Case Design**   Extract testable requirements from the requirement specifications and design test conditions which accurately reflect the functional enhancements and changes.  [Add design elements/requirements specific to this test activity].  Identify test approach about test case design before identify the detail of test case test script in next step. To declare what is the concern item of each module or functional/non functional area and what the test case design is for detect the defect. The test technique should be state for this section. XYZ bank test coverage matrix should be mention and state that how to ensure the project will follow this guideline.  Test design should mention about the group of regression impact. If the most of the defect occur on the area which test case design should be re-test. And the regression test approach should be mention as well.   * 1. **Test Scheduling**   Identify test schedule for <Test level> base on <project master plan version>. The high level plan for testing preparation, test execution, defect fixing, and data test preparation period are identified as below. <The format of schedule can demonstrate in structure of table of calendar or schedule in Microsoft project>  [For complex test schedules reference any external tools or systems you will be using/developing to support the scheduling. Include specific flags or nomenclature that will be used to identify/classify test scripts and test cases that have to be executed at a specific point in the schedule. (pre or post batch, day one, after script xyz has executed, etc]  Tables?   * 1. **Data Build**   Identify plan to get data for each testing area. Mention the approach and activity plan for getting test data properly. The process step of prepare data mention here.  [Document the strategies that will be used to generate or extract appropriate data for the test effort].   * 1. **Results/Sign-off**  1. **Suspension/Resumption Criteria**   Testing will halt for a particular project item (or function) when:  • A critical problem is identified and where the potential code fix will require substantial re-testing of that function  • It is identified that the business or technical specifications require major modifications due to escalated test issues and those modifications would require additional test analysis and or modification to the Detailed Test Plan.  • The test regions or test environment are not available (for any reason).  • The test regions or test environment suffer performance problems below 50% of their normal operating capacity, such that a region fix will require substantial re-testing of that function.  • [Document all other suspension/resumption. Make sure the resumption criteria are unambiguously defined].  <Select from below table only testing level related on this DTP> consider the entry and Exit criteria reasonable on your project>   1. **Pass/Fail Criteria**   The specific pass/fail criteria for the testing at both the test cycle and release level are identifying in table below. This can include percentage of severity 3 and 4 defects that will be allowed to migrate between test/production environment and any specific business defined criteria. Identify information in this part , select only current testing level involve in this DTP.  [Document any specific pass/fail criteria for the testing at both the test cycle and the overall test activities. This can include percentage of severity 3 and 4 defects that will be allowed to migrate between test/production regions and any specific business defined criteria].   1. **Suspension Criteria**   In this section, we specify when to stop the testing.  Example: If any of the major functionalities are not functional or system experiences login issues then testing should suspend.  Table? |

| 4.0 Test conditions |
| --- |
| * 1. … |

| 1. Test Environments   List and/or graphically show the proposed testing environment/s. For small testing projects complete all of the listed sections. Larger projects may require a separate Test Environment Plan to be produced. If Master test plan already specify may refer to MTP plan instead. |
| --- |
| * 1. **Client Side Infrastructure**   Provide detailed list/s of required hardware and software at Client side to support testing activity of each test level and test environment. May use tabular format for explain the content.   * 1. **Host/Server Side Infrastructure**   Provide detailed list/s of required hardware and software at Host or Server to support testing activity of each test level and test environment. May use tabular format for explain the content.     * 1. **Middleware**   Provide detailed list/s of required Middleware to support testing activity of each test level and test environment. Eg. Test engine or test stub required for interface test.   * 1. **Test Data Preparation**   Define the data subset/s that needs to be pre-loaded into the test environments. This content in is mention about infrastructure preparation view, it does not about the data condition or concept that mention in section 3.2 Data test.  [Provide detailed data requirements. Where specific data is required to establish a data condition or execute a test script provide either a list of these data requirements or reference the location of this information. Include the processes that will be used to deliver the data to the test region. (production extract by operations staff, build by automated scripts, generated by the development team/vendor etc)]. |

| 10.0 testing tasks  In this section, we specify the list of testing tasks we need to complete in the current project.  Example: Test environment should be ready prior to test execution phase. Test summary report needs to be prepared. |
| --- |
| * 1. **User Interfaces**   The logic behind the interactions between the users and the software. This includes the sample screen layout, buttons and functions that would appear on every screen, messages to be displayed on each screen and the style guides to be used.   * 1. **Hardware Interfaces**   All the hardware-software interactions with the list of supported devices on which the software is intended to run on, the network requirements along with the list of communication protocols to be used.   * 1. **Communications Interfaces**   Determination of all the communication standards to be utilized by the software as a part of the project.   * 1. **Software Interfaces**   The interaction of the software to be developed with other software components such as frontend and the backend framework to the used, the database management system and libraries describing the need and the purpose behind each of them. |

| 11.0 Environmental needs  List of hardware, software and any other tools that are needed for a test environment. |
| --- |
| * 1. **User Interfaces**   The logic behind the interactions between the users and the software. This includes the sample screen layout, buttons and functions that would appear on every screen, messages to be displayed on each screen and the style guides to be used.   * 1. **Hardware Interfaces**   All the hardware-software interactions with the list of supported devices on which the software is intended to run on, the network requirements along with the list of communication protocols to be used.   * 1. **Communications Interfaces**   Determination of all the communication standards to be utilized by the software as a part of the project.   * 1. **Software Interfaces**   The interaction of the software to be developed with other software components such as frontend and the backend framework to the used, the database management system and libraries describing the need and the purpose behind each of them. |

| 12.0 Responsibilities  We specify the list of roles and responsibilities of each test tasks.  Example: Test plan should be prepared by Test Lead. Preparation and execution of tests should be carried out by testers. |
| --- |
| * 1. **User Interfaces**   The logic behind the interactions between the users and the software. This includes the sample screen layout, buttons and functions that would appear on every screen, messages to be displayed on each screen and the style guides to be used.   * 1. **Hardware Interfaces**   All the hardware-software interactions with the list of supported devices on which the software is intended to run on, the network requirements along with the list of communication protocols to be used.   * 1. **Communications Interfaces**   Determination of all the communication standards to be utilized by the software as a part of the project.   * 1. **Software Interfaces**   The interaction of the software to be developed with other software components such as frontend and the backend framework to the used, the database management system and libraries describing the need and the purpose behind each of them. |

| 13.0 Staffing and training needs  Plan training course to improve the skills of resources in the project to achieve the desired goals. |
| --- |
| * 1. **User Interfaces**   The logic behind the interactions between the users and the software. This includes the sample screen layout, buttons and functions that would appear on every screen, messages to be displayed on each screen and the style guides to be used.   * 1. **Hardware Interfaces**   All the hardware-software interactions with the list of supported devices on which the software is intended to run on, the network requirements along with the list of communication protocols to be used.   * 1. **Communications Interfaces**   Determination of all the communication standards to be utilized by the software as a part of the project.   * 1. **Software Interfaces**   The interaction of the software to be developed with other software components such as frontend and the backend framework to the used, the database management system and libraries describing the need and the purpose behind each of them. |

| 14.0 Schedule  Complete details on when to start, finish and how much time each task should take place.  Example: Perform test execution – 120 man-hours, Test Reporting – 30 man-hours |
| --- |
| * 1. **User Interfaces**   The logic behind the interactions between the users and the software. This includes the sample screen layout, buttons and functions that would appear on every screen, messages to be displayed on each screen and the style guides to be used.   * 1. **Hardware Interfaces**   All the hardware-software interactions with the list of supported devices on which the software is intended to run on, the network requirements along with the list of communication protocols to be used.   * 1. **Communications Interfaces**   Determination of all the communication standards to be utilized by the software as a part of the project.   * 1. **Software Interfaces**   The interaction of the software to be developed with other software components such as frontend and the backend framework to the used, the database management system and libraries describing the need and the purpose behind each of them. |

| 15.0 Risks and contingencies  In this section, we specify the probability of risks and contingencies to overcome those risks.  Example: Risk – In case of a wrong budget estimation, the cost may overrun. Contingency Plan – Establish the scope before beginning the testing tasks and pay attention in the project planning and also track the budget estimates constantly. |
| --- |
| * 1. **User Interfaces**   The logic behind the interactions between the users and the software. This includes the sample screen layout, buttons and functions that would appear on every screen, messages to be displayed on each screen and the style guides to be used.   * 1. **Hardware Interfaces**   All the hardware-software interactions with the list of supported devices on which the software is intended to run on, the network requirements along with the list of communication protocols to be used.   * 1. **Communications Interfaces**   Determination of all the communication standards to be utilized by the software as a part of the project.   * 1. **Software Interfaces**   The interaction of the software to be developed with other software components such as frontend and the backend framework to the used, the database management system and libraries describing the need and the purpose behind each of them. |