|  |
| --- |
| Team PHarmacon  Master Test Plan |

Version information

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Remarks** | **Author** |
| **V1.0** | 11/03/2018 | Initial Version | Ryan Smith & Jette McKellar |

Management summary

|  |  |  |  |
| --- | --- | --- | --- |
| **Project objective**  Team Pharmacon has been discovered a need for an application that aids hospital pharmacist in the tracking of medicinal errors and wastage. The application will be called Pharmacy Error Tracker (P.E.T.). The error tracking will be done with a web interface that is designed to be used across any platform. It will also allow hospital pharmacists to see visual representations of all the tracked errors, helping them track down deficiencies in instructions, routines and training. | | | |
| **Test objective**  The objective of testing is ensuring that each component of the application works as expected. | | | |
| **Short description of the test approach**  We will be conducting unit tests and integration tests to test the system as development proceeds. We will be conducting System Tests and User Acceptance tests before entering the Transition phase of the project. During the Transition phase, the deployment of the project will be treated as a beta test. | | | |
| **Results to be realized** | | | |
| *Result*   * Well executed unit tests * Well executed and finished integration test * Well executed and finished system test * Well executed and finished user acceptance test * Well executed and finished beta test report | | *Document*   * Unit Test report * Integration Test report * System Test Report * UAT Test Report * Beta Test Report | *Delivery date* 3 September 2018  3 September 2018  3 September 2018  3 September 2018  3 October 2018 |
| **Qualitative objectives**  Each test level needs to be completed on time as development proceeds. The test reports need to clearly demonstrate that each system object meets the acceptance criteria. | | | |
| **Test process risks and measures**  *Test process risks*  • Team members may fail to properly report on unit and integration tests done. | *Measures to be taken*  • Test reports are to be added as work items to each iteration plan. | | |
| **Go/no-go decisions**  After each test level the team member responsible will make sure that a test report is drawn up. This report will, after review with the other team members and the subject lecturer, be used to decide if the project is capable of proceeding as planned. | | | |

Table of Contents

1 Introduction 1

1.1 Project and project objective 1

1.2 Objective of the master test plan 1

1.3 Involved in creating the master test plan 1

2 Assignment formulation 2

2.1 Client 2

2.2 Supplier 2

2.3 Assignment 2

2.4 Scope 2

2.4.1 Within scope 2

2.4.2 Out of scope 2

2.5 Preconditions and assumptions 3

2.6 Acceptants and acceptance criteria 3

2.6.1 Acceptants 4

2.6.2 Acceptation criteria 4

3 DocumentatiON 5

3.1 Basis for the master test plan 5

3.2 Standards 5

3.3 Test basis 5

4 Test strategy 6

4.1 Product risk analyses 6

Risk table 7

4.2 Test strategy 7

5 Approach 9

5.1 Test levels 9

5.2 Evaluation 9

5.3 The <name test level> 9

5.3.1 Goal 9

5.3.2 Short description 9

5.3.3 Responsible 9

5.3.4 << Optional: test environment to be used >> 10

5.4 Phasing per test level 10

5.5 Test products 11

5.6 Review plan 11

5.7 Entrance and exit criteria for each test level 12

5.7.1 << Optional: Functional Acceptance Test >> 12

5.7.2 << Optional: User Acceptance Test >> 12

5.8 Go/No go 12

6 Organization 14

6.1 Organization structure 14

6.2 Roles, tasks and responsibilities 14

6.2.1 << Optional: Trainings- and coaching’s necessity >> 14

6.3 Structure of meetings 14

6.4 Structure of reporting 15

6.5 Completion 15

7 Infrastructure 16

7.1 Test environments 16

7.2 Test tools 16

7.3 Office setup 17

8 Management 18

8.1 Test process management 18

8.2 Test infrastructure management 18

8.3 Test product management 18

8.4 Defects procedure 18

9 Test process risks and countermeasures 19

10 Global Estimation & Planning 20

10.1 Estimation 20

10.2 Planning 20

10.3 Milestones 21

11 Glossary 22

# Introduction

## Project and project objective

This master test plan fits to the project plan for Pharmacy Error Tracker. The Pharmacy Error Tracker project objective is to create a webpage application that pharmacists can use to track the occurrences of procedural errors in their prescriptions.

## Objective of the master test plan

The objective of the Master Test Plan (MTP) is to inform all who are involved in the test process about the approach, the activities, and the end products to be delivered for the Pharmacy Error Tracker project tests.

The master test plan describes this approach, the activities and the end products, which will need further elaboration in separate system test plans. These system test plans need to be abstracted from this master test plan.

## Involved in creating the master test plan

|  |  |
| --- | --- |
| **Name** | **Responsibility** |
| Ryan Smith | Write MTP |
| Jette McKellar | Write MTP |
| Leonard Meerwood | <Approve MTP> |

# Assignment formulation

## Client

The client is potentially hospital pharmacies that require an application to track errors that occur during the distribution of medication.

## Supplier

Supplier is Team Pharmacon, the developers of the Pharmacy Error Tracker.

## Scope

### Within scope

The single page web application will enable a user to add errors that occur in the dispensing of drugs to patients. Depending on the user’s security level, they will be able to run different visualisations (graphs, charts, and Excel reports) to see causes of errors in a simple way, complete searches/filtering of records as well as sending various reports to a contact or contact list. There will also be an ability to add and delete users and delete error records.

Users will need to be able to:

* Add an error, selecting the appropriate details from various dropdown menus. User will also be able to enter a short free text description of circumstances leading up to error.
* Search/filter errors in the database.
* Amend an error, correcting false information entered and saved by accident.
* Export the data concerning the entered errors to Excel spreadsheet filetype.
* Manage contacts, e.g., add, update or delete contacts.
* Send reports to a contact or a contact list.
* Create various visualisations (charts, graphs) from the data to enable easier interpretation.
* Manage users, e.g., add, update or delete users.
* Delete an error form.
* Customise the error form.

There will be three levels of authority within this application:

* General user – will be able to:
  + Enter and amend errors
  + Search/filter errors
* Supervisor – will be able to:
  + Enter and amend errors
  + Search/filter errors
  + Generate reports and graphs
  + Send reports to a contact or a contact list
  + Manage contacts (add, update, or delete)
* Administrator – will be able to:
  + Enter and amend errors
  + Search/filter errors
  + Generate reports and graphs
  + Send reports to a contact or a contact list
  + Manage contacts (add, update, or delete)
  + Manage users (add, update, or delete)
  + Customise form

Tests will need to be carried out to ensure:

* Error data entered is saved and stored in a database.
* Updating an error record will update the correct record in the database.
* Searches/filtering of the database will return all records that relate to that specific search/filter.
* When a graph, chart, or visualisation is requested, all data relating to that specific visualisation is extracted from the database to ensure a meaningful visualisation is created. The visualisation should be downloadable to the local machine on which the application is running.
* When extracting data to create an Excel spreadsheet report, all appropriate data relating to the report is included. The report is to be downloadable to the local machine on which the application is running.
* When a report is to be sent to a contact or contact list, an appropriate email application is opened with the report added as attachments to the message of the email. A subject line is to be added to the email giving the report name.
* When a new contact is added to the contacts, their data is saved to the correct table within the database and it is possible to send emails to the contact.
* Various contacts can be grouped together to form contact lists to enable reports to be sent to a contact list rather than individuals if required.
* Updating of contact information saves the new information to the correct record in the database and is recalled when a new email is sent to that updated contact.
* Deleting a contact from the contact list will make the record unavailable to authorised users.
* A newly added user can only perform tasks as outlined by the authorisation level given to the user (user, supervisor, administrator).
* When a user logs into the application, they will see the options available to them as per their authorisation level.
* When a user changes their password, they can login to application at their authorised level.
* An update of user preferences ensures user can only perform tasks outlined by the authorisation level given to the user (user, supervisor, administrator).
* A deleted user will not have access to the application. However, user details are to remain in the database for auditing purposes and to appear on reports as required.
* A form that is customised will still enable data to be saved to appropriate database tables, enabling data to be extracted in same method used prior to changes to form being made.

# Test strategy

The time available for testing is limited; not everything can be tested with equal thoroughness. This means that choices have to be made regarding the depth of testing. Also it is strived to divide test capacity as effective and efficient as possible over the total test project. This principle is the basis of the test strategy.

The test strategy is based on risks: a system has to function in practice to an extent that no unacceptable risks for the organization arise from it. If the delivery of a system brings along many risks, thorough testing needs to be put in place; the opposite of the spectrum is also true: 'no risk, no test'.

The first step in determining the test strategy is the execution of a *product* *risk analyses*. This is elaborated in §4.1.

The test strategy is subsequently based on the results of the risk analyses. The test strategy lays down *what,* *how* and *when* (in which test level) is being tested and is focused in finding the most important defects as early as possible for the lowest costs. This can be summarized as testing with an optimal use of the available capacity and time. The test strategy is described in §4.2.

## Product risk analyses

The product risks can be found in the [Pharmacy Error Risk List document](https://bitbucket.org/itc303teampharmacon/pharmacy_app/src/e5fec6ae6faacaddb98a8051638e783cbfe42f44/documents/Pharmacy%20Error%20Tracker%20Risk%20List.xlsx?at=master&fileviewer=file-view-default).

## Test strategy

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Characteristic /object part** | **Unit test** | **Implementation** | **System Test** | **User acceptance test** |
| “Add an Error” | * Error data is saved to the correct fields within the database | By 27/05/2018 |  | By 31/05/2018 |
| "Add a User"  "Produce Visualisations"  "Modify Error in System"  "Output Error Data to Excel" | * A newly added user can only perform tasks as outlined by the authorisation level given to the user (user, supervisor, administrator). * When a graph, chart, or visualisation is requested, all data relating to that specific visualisation is extracted from the database to ensure a meaningful visualisation is created. * The visualisation should be downloadable to the local machine on which the application is running. * Updating an error record will update the correct record in the database. * When extracting data to create an Excel spreadsheet report, all appropriate data relating to the report is included. The report is to be downloadable to the local machine on which the application is running. | By 16/07/2018 |  | By 20/07/2018 |
| "Log In"  "Add a Contact"  "Send Report to Contacts"  "Edit a Contact" | * When a user logs into the application, they will see the options available to them as per their authorisation level. * When a new contact is added to the contacts, their data is saved to the correct table within the database. * It is possible to send emails to the contact. * Various contacts can be grouped together to form contact lists to enable reports to be sent to a contact list rather than individuals if required. * When a report is to be sent to a contact or contact list, an appropriate email application is opened with the report added as attachments to the message of the email. * A subject line is to be added to the email giving the report name. * Updating of contact information saves the new information to the correct record in the database and is recalled when a new email is sent to that updated contact. | By 30/07/2018 |  | By 3/08/2018 |
| "Manage User Details – Preferences"  “Delete a User"  "Change Password"  "Edit Error Submission Form"  "Remove a Contact" | * An update of user preferences ensures user can only perform tasks outlined by the authorisation level given to the user (user, supervisor, administrator). * A deleted user will not have access to the application. * User details are to remain in the database for auditing purposes and to appear on reports as required. * When a user changes their password, they can login to application at their authorised level. * A form that is customised will still enable data to be saved to appropriate database tables, enabling data to be extracted in same method used prior to changes to form being made. * Deleting a contact from the contact list will make the record unavailable to authorised users | By 13/08/2018 |  | By 17/08/2018 |
| Deploy Application in Trial Environment – External UAT Testing Round 1 |  | By 10/09/2018 | Test first half of application to ensure the application works. Go through use case testing. | By 14/09/2018 |
| Deploy Application in Trial Environment – External UAT Testing Round 2 |  | By 24/09/2018 | Test second half of application to ensure the application works. Go through use case testing. | By 28/09/2018 |

# Approach

In this chapter each test level in the test strategy (the *what*) will be translated to a concrete test approach (the *how*). << Make sure that the described test approach reflects the test strategy from chapter 4! Each element from the test strategy has to return here! This paragraph can be more concise if there will be test plans (TP) drawn up for each test level. (Refer to the TP’s that have to be written). There are two important factors that determine whether TP’s are being written or not:

* The size of the project;
* The level of uncertainties and ambiguities that are there at the moment of writing the MTP.>>

## Test levels

For this MTP the following test levels are acknowledged:

|  |  |
| --- | --- |
| **Test level** | **Goal** |
| Unit Test | To ensure that each part of the developed code works as intended and is fit for purpose. |
| Integration Test | To ensure that each part of the developed code works as intended when introduced to the rest of the project, and that interactions between sections of code produce the correct results. |
| System Test | To ensure that the completed system works as expected once each aspect of the code is fully integrated. |
| User Acceptance Test | To ensure that the completed system is capable of correctly executing each use case as described in the use case description. |
| Beta test | To ensure that the completed product works as intended in the production environment, when being used by actual users. |

Each Iteration Plan created by Team Pharmacon will evaluate the project strategy and status. These documents can be located in the [Iteration Folder](https://bitbucket.org/itc303teampharmacon/pharmacy_app/src/e5fec6ae6faacaddb98a8051638e783cbfe42f44/documents/Iteration%20Plan/?at=master).

## Test Levels

### Unit Tests

4.2.1.1: Goal

To ensure that each part of the developed code works as intended and is fit for purpose.

4.2.1.2 Short Description

Unit tests will ensure that each coded function works as intended in isolation. The coder will specify and execute this test level, using the development environment NodeJS.

4.2.1.3 Who Is Responsible

The coder is responsible for this level of test.

4.2.1.4 Test Environment

The Unit Tests will be executed within NodeJS.

### Integration Tests

4.2.2.1: Goal

To ensure that each part of the developed code works as intended when introduced to the rest of the project, and that interactions between sections of code produce the correct results.

4.2.2.2 Short Description

Integration tests will ensure that each coded fuction works as intended when integrated with the rest of the code, by examining interactions between parts of code and testing that the results of those interactions are correct. The coder will specify and execute this test level, using the development environment NodeJS.

4.2.2.3 Who Is Responsible

The coder is responsible for this level of test.

4.2.2.4 Test Environment

The Integration Tests will be executed within NodeJS.

### System tests

4.2.3.1: Goal

To ensure that the completed system works as expected once each aspect of the code is fully integrated.

4.2.3.2 Short Description

System tests will ensure that the entire system works as functions are added. The coder will specify and execute this test level, using the production environment of the two common web browsers Mozilla Firefox and Google Chrome.

4.2.3.3 Who Is Responsible

The coder is responsible for this level of test.

4.2.3.4 Test Environment

The System Tests will be executed within the web browsers Mozilla Firefox and Google Chrome.

### User Acceptance Tests

4.2.4.1: Goal

To ensure that the completed system is capable of correctly executing each use case as described in the use case description.

4.2.4.2 Short Description

User Acceptance tests will ensure that the system is capable of performing each use case correctly. The project team will specify and execute this test level, using the production environment of the two common web browsers Mozilla Firefox and Google Chrome.

4.2.4.3 Who Is Responsible

The project team is responsible for this level of test.

4.2.4.4 Test Environment

The User Acceptance Tests will be executed within the web browsers Mozilla Firefox and Google Chrome.

### Beta Test

4.2.5.1: Goal

To ensure that the completed product works as intended in the production environment, when being used by actual users.

4.2.5.2 Short Description

Beta tests will ensure that the system is capable of performing correctly in the production environment. The project team will specify this test level, and the interested party will execute it, using the production environment of the user’s preferred web browser/s.

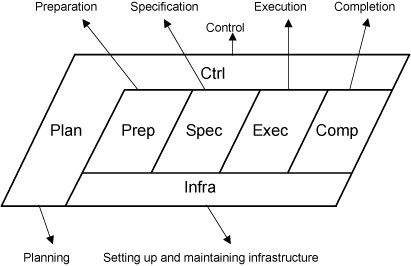
4.2.5.3 Who Is Responsible

The project team is responsible for this level of test.

4.2.5.4 Test Environment

The User Acceptance Tests will be executed within the web browsers Mozilla Firefox and Google Chrome.

## Phasing per test level



In the **Planning** phase, the test manager formulates a coherent approach that is supported by the client to adequately execute the test assignment. This is laid down in the test plan. In the **Control** phase the activities in the test plan are executed, monitored, and adjusted if necessary. The **Setting up and maintaining infrastructure** phase aims to provide the required test infrastructure that is used in the various TMap phases and activities. The **Preparation** phase aims to have access to a test basis, agreed with the client of the test, of adequate quality to design the test cases. The tests are specified in the Specification phase and executed in the **Execution** phase. This provides insight into the quality of the test object. The test assignment is concluded in the **Completion** phase. This phase offers the opportunity to learn lessons from experiences gained in the project. Furthermore, activities are executed to guarantee reuse of products.

| **Deliverable** | **Authors** | **Type review** | **Reviewers** |
| --- | --- | --- | --- |
| Master Test Plan | Ryan, Jette | LCOM | James Tulip |
| Deploy Executable Architecture in Trial Environment | Team Pharmacon | LCAM | James Tulip |
| Deliver Initial Operation Capability Milestone | Team Pharmacon | IOCM | James Tulip |
| Deliver Product Release Milestone | Team Pharmacon | PRM | James Tulip |