# Pharmacy Error Tracker

# Use-Case: Add an error

1. Brief Description

When a user

Wants to add an error they enter the error details into the app

So that the error can be stored and reported on later.

1. Actors
   1. Basic user

* A basic user of the system, with no additional permissions

1. Pre-Conditions
   1. An error has occurred in the pharmacy processes that needs to be recorded.
2. Normal Flow

The use case begins when the user navigates to the screen where new error records are added.

|  |  |
| --- | --- |
| Actor | System |
| 1.1: Enters the details of the errors into the various fields. | 2.1: The system checks the details are valid (inputs are sanitised) |
|  | 2.2: The system adds a record to its internal SQLLite database |
|  | 2.3: The system sends the record to the externally hosted database. |
|  | 2.4: The system displays a success message to the user. |

The use case ends.

1. Alternate Flows
   1. Failure flow 1

If at step 2.1 of the normal flow the inputs are invalid, then:

|  |  |
| --- | --- |
| Actor | System |
|  | 2.1.1: The system displays a failure message to the user, indicating the problem field. |

* 1. Failure flow 2

If at step 2.2 of the normal flow the inputs are invalid, then:

|  |  |
| --- | --- |
| Actor | System |
|  | 2.2.1: The system displays a failure message to the user, indicating the problem. |

* 1. Sync failure

If at step 2.3 of the normal flow the local database fails to connect to the remote database, then:

|  |  |
| --- | --- |
| Actor | System |
|  | 2.3.1: The system displays a notice, indicating the problem, and continues to attempt to connect in the background. |

1. Subflows
   1. <subflow 1>
2. Key Scenarios
   1. <scenario 1>

<brief description>

1. Post-conditions
   1. The record of the error should be created and should be synced with the desktop server.
2. Special Requirements
   1. <special requirement 1>