

1. Home .....	4
1.1 CIDA Functional Requirements .....	5
1.1.1 CIDA GUI Mockups .....	24
1.1.1.1 GM102 Welcome Page for StudyWorker User .....	24
1.1.1.2 GM103 Welcome Page for ReportsWorker .....	25
1.1.1.3 GM104 Add New User (CIDA) .....	26
1.1.1.4 GM301 Add Participant to Study .....	26
1.1.1.5 GM401 Remove Participant from Study .....	28
1.1.1.6 GM701 Study Alerts .....	29
1.1.1.7 GM801 Scheduled Appointments and Attendance Report .....	29
1.1.2 CIDA Test Cases .....	30
1.1.2.1 TC104 Add a new user (roles) .....	30
1.1.2.2 TC201 Send Request for Campaign list .....	31
1.1.2.3 TC301 Add a Participant to Study .....	32
1.1.2.4 TC302 Randomise participants .....	33
1.1.2.5 TC303 Send 'Add Recipient to Campaign' Request to Mobilisr .....	34
1.1.2.6 TC401 Remove Participant from Study .....	35
1.1.2.7 TC402 Send 'Remove Recipient from Campaign' Request to Mobilisr .....	36
1.1.2.8 TC504 Send 'Just Send SMS' Request to Mobilisr .....	37
1.1.2.9 TC601 - Issue an appointment date for a new patient .....	38
1.1.2.10 TC602 - Update an existing patient's appointment date .....	39
1.1.2.11 TC603 - Dispense directly to a patient who has been on an appointment .....	39
1.1.2.12 TC604 - Dispense to an existing patient for a later pickup .....	40
1.1.3 CIDA Use Cases .....	41
1.1.3.1 UC101 Study worker Welcome page .....	41
1.1.3.2 UC102 Welcome screen for Reports Worker .....	41
1.1.3.3 UC104 Add new user .....	42
1.1.3.4 UC601 - Issue an appointment date for a new patient .....	43
1.1.3.5 UC602 - Update an existing patient's appointment date .....	44
1.1.3.6 UC603 - Dispense directly to a patient who has been on an appointment .....	45
1.1.3.7 UC604 - Dispense to an existing patient for a later pickup .....	46
1.1.3.8 UC301 Add Participant To Study .....	47
1.1.3.9 UC401 Remove Participant From Study .....	47
1.1.4 CIDA Requirement Descriptions .....	49
1.1.4.1 FR 601 - Issue an appointment date for a new patient .....	49
1.1.4.2 FR602 - Update an existing patient's appointment date .....	49
1.1.4.3 FR 603 - Dispense directly to a patient who has been on an appointment .....	49
1.1.4.4 FR604 - Dispense to an existing patient for a later pickup .....	50
1.1.4.5 FR301 Add Participant to Study .....	50
1.1.4.6 FR401 Remove Participant from Study .....	51
1.1.5 Mind Map - CIDA Functional Requirements .....	51
1.2 Clinical Dispensing .....	51
1.3 Database Backup .....	54
1.4 Functional Requirements List (v3.5X, Rewrite & CIDA) .....	55
1.5 Getting started .....	57
1.5.1 Postgresql .....	60

1.5.2 Troubleshooting .....	60
1.6 How to contribute .....	63
1.7 iDART 3.5 Functional Requirements .....	63
1.7.1 iDART GUI Mockups .....	67
1.7.1.1 GM101 Add Update Clinic .....	67
1.8 iDART CIDA Functionality .....	68
1.9 iDART CIDA Technical Doc .....	69
1.10 iDART Encore Functional Requirements .....	70
1.11 iDART Feature Stories .....	95
1.12 iDART GUI Map .....	96
1.13 iDART Integration .....	97
1.14 iDART Managed Entities .....	98
1.15 iDART Reports .....	99
1.16 iDART Roadmap .....	101
1.16.1 Change Log .....	101
1.17 iDART Test Cases .....	108
1.17.1 Monthly Stock Receipts report .....	109
1.17.2 TC93X - Test Data for Package Movement Reports (downReferralModel = offline) .....	110
1.17.3 TC93X - Test Data for Package Movement Reports (downReferralModel = online - NOT DONE) .....	111
1.17.4 TC911 - Patient History Report .....	113
1.17.5 TC912 - Episodes Started Report .....	114
1.17.6 TC913 - Episodes Ended Report .....	116
1.17.7 TC914 - Package Tracking Report .....	116
1.17.8 TC921 - Monthly Receipts and Issues Report .....	117
1.17.9 TC926 - ARV Drug Usage .....	118
1.17.10 TC931 - Packages Created .....	119
1.17.11 TC932 - Packages Leaving Pharmacy .....	119
1.17.12 TC933 - Packages Received at Clinic .....	120
1.17.13 TC934 - Packages Collected .....	121
1.17.14 TC935 - Packages Awaiting Pickup .....	121
1.17.15 TC936 - Patients Expected On A Day .....	122
1.17.16 TC937 - Missed Appointments Report .....	123
1.17.17 TC941 - Drug Combinations (CSV) .....	123
1.17.18 TC944 - Prescribing Doctors Report .....	124
1.17.19 TC945 - PEPFAR Report .....	125
1.17.20 TC946 - Clinic Indicators Report .....	126
1.17.21 TC947 - Clinic Indicator Report (Draft) .....	127
1.17.22 Test Cases to be completed .....	128
1.17.23 Updating from iDART 3.0 to latest .....	128
1.18 Notes .....	128
1.19 Notes on creating a "CIDA" type connection to Communicate .....	129
1.20 Old wiki archive .....	130
1.20.1 checklists .....	130
1.20.2 conventions .....	133
1.20.3 dbscripts .....	138

1.20.4 hardware .....	140
1.20.5 ideas_concurrency .....	141
1.20.6 labelTesting .....	147
1.20.7 Itsp .....	148
1.20.8 network .....	151
1.20.9 svn_merges .....	152
1.20.10 user_docs .....	156
1.21 Release process .....	157
1.22 Screens .....	158
1.22.1 Clinic .....	158
1.22.2 Delete Stock/Prescriptions/Packages .....	159
1.22.3 Destory Stock .....	161
1.22.4 Patient Packaging .....	162
1.22.5 Stock Center .....	164
1.23 SPRINT Meeting Notes .....	167
1.23.1 18 May 2010 .....	167
1.23.2 01 June 2010 .....	167
1.23.3 29 June 2010 .....	168
1.23.4 13 July 2010 .....	169
1.23.5 09 Sept 2010 .....	170
1.24 Wish List .....	171

# Home

## Welcome to the iDART Wiki!

iDART is a software solution designed to support the dispensing of ARV drugs in the public health care sector. It supports pharmacists in their important role of dispensing accurately to an increasing number of patients whilst still being able to engage and assist the patient.

iDART is a product of Cell-Life.

[Download latest release](#) which includes a user guide and quick reference guide.

## Getting Started

Check out our getting started guide to learn how to work with the iDART source code. Developers should sign up to the mailing lists:

- [Developers mailing list](#) (archive ([shttp://sourceforge.net/mailarchive/forum.php?forum\\_name=idart-developers](http://sourceforge.net/mailarchive/forum.php?forum_name=idart-developers)))  
The developers list is where discussions relating to iDART development happen. If you intend to do any iDART development it would be a good idea to join this list.
- [Ticket mailing list](#)(archive)  
All changes to iDART tickets are copied to this mailing list. This is useful if you are want to keep abreast of ticket changes. (Tickets are used to keep track of what work is being done or needs to be done on iDART).Submitting Issues

Submit bug reports, feature requests and any other issues at the [issue tracker](#)

## Contributing Code

[How to contribute code to iDART](#)

## License

iDART is an open source project and is available under the [GPL v2.0 license](#).

### *Navigate space*



## Recently Updated

- [Notes on creating a "CIDA" type connection to Communicate](#)  
Jul 10, 2014 • updated by Diné Bennett • [view change](#)
- [Postgresql](#)  
Jan 30, 2014 • updated by Dagmar Timler • [view change](#)
- [Wish List](#)  
Jan 27, 2014 • updated by sikander abdul karim noor mahomed • [view change](#)
- [destroy\\_stock.png](#)  
Nov 05, 2012 • attached by Kevin Sewell
- [Destory Stock](#)  
Nov 05, 2012 • created by Kevin Sewell
- [Delete Stock/Prescriptions/Packages](#)  
Oct 23, 2012 • updated by Kevin Sewell • [view change](#)
- [delete\\_prescription.png](#)  
Oct 23, 2012 • attached by Kevin Sewell
- [delete\\_stock.png](#)  
Oct 23, 2012 • attached by Kevin Sewell
- [delete\\_package\\_item.png](#)  
Oct 23, 2012 • attached by Kevin Sewell

-  [delete\\_package.png](#)  
Oct 23, 2012 • attached by Kevin Sewell
-  [Patient Packaging](#)  
Oct 12, 2012 • updated by Kevin Sewell • view change
-  [Stock Center](#)  
Oct 12, 2012 • created by Kevin Sewell
-  [add\\_pharmacy.png](#)  
Oct 12, 2012 • attached by Kevin Sewell
-  [update\\_pharmacy.png](#)  
Oct 12, 2012 • attached by Kevin Sewell
-  [update\\_facility.png](#)  
Oct 12, 2012 • attached by Kevin Sewell

## CIDA Functional Requirements

| *CIDA Functional Requirements - Table of Contents*

- General (FR100)
- Campaign List (FR200)
- Add Participant to Study (FR300)
- Remove Participant from Study (FR400)
- Reminder Module (FR500)
- **Appointments (FR600)**
- **Study Alerts (FR700)**
- **Reports (FR800)**
- Parking Lot (including notes for EMIT & Mobilisr requirements)

[Link to Mindmap - CIDA Functional Requirements](#)

### General (FR100)

Req No	Req Name	Requirement Description (includes Business Rules)	Process Diagram	GUI Mockup	Use Case	Test Case		Design (BA)	Dev (Dev)	Test (BA/Dev )
FR101	Welcome Page for StudyWorker	The studyWorker user needs access to Update Existing Patient, Add Participant to Study, Remove Participant from Study & Logoff	N.A.	GM102	UC101	ba to test				
FR102	Welcome Page for Reports Worker	The reports Worker user needs access to Reports & Logoff	N.A.	GM103	UC102	ba to test				

FR103	Modular Access for CIDA Study	Create a property file called CIDA, defaulted to false. The functionality developed for this release is generally specifically for the research study but we don't want a separate version completely (for managing bug fixes, etc.) Setting this property to true would: * give permissions to users, * others (as we identify them)	N.A.	N.A.		ba to test					
FR104	Add New iDART User (CIDA)	If the CIDA property is set to TRUE, then implement the redesigned screens for 'Add		[GM104	UC104	TC104					

a New iDART user'. Note that the low level users (studyWorker and reports Worker) will not have access to the user screen (for security reasons we don't want them to be able to change their permissions and give themselves pharmac y access, nor do we want them to add new users). As a result of this, it will only be possible to change these low-level users by manipulating the database directly. This is not ideal, but is a

quick fix  
for the  
CIDA  
project.  
Only  
pharmac  
y staff  
type of  
users  
will be  
able to  
add the  
low-level  
users to  
**the syst**  
em.

Each  
user has  
to  
belong  
to a user  
role,  
either a  
pharmac  
ist, a  
study  
worker

		or a report worker.								

### Campaign List (FR200)

Req No	Req Name	Requirement Description (includes Business Rules)	Process Diagram	GUI Mockup	Use Case	Test Case		Design (BA)	Dev (Dev)	Test (BA/Dev)
FR201	Send Request for Campaign List	Allow the user to send a request for the campaign list that belongs to the organisation. The user will require a Mobilisr Organisation name as well as a Mobilisr Password. This will be accessible through a 'Refresh Campaign List' link / button on the GUI for FR301 (Add Participant to Study). The list will be fetched once, and then only upon a request by the user through	NA	NA	NA	TC201				

the  
'Refresh'  
link. The  
response  
would be  
a list of  
campaign  
s which  
will be  
used to  
populate  
the drop  
down list.  
Refer to  
[REST](#)  
API  
Requirem  
ents  
(API001  
Get

		Campaigns)							
--	--	------------	--	--	--	--	--	--	--

### Add Participant to Study (FR300)

Req No	Req Name	Requirement Description (includes Business Rules)	Process Diagram	GUI Mockup	Use Case	Test Case		Design (BA)	Dev (Dev)	Test (BA/Dev)
FR301	Add Participant to Study	A user with study permission must be able to add a participant to the study. This includes capturing the mobile number, preferred language, informed consent, etc. update this requirement - if the cellphone number is different to the one that was already saved to the patient, inform the user and overwrite it	NA	GM301	UC301	TC301			MS	

FR302	Randomise Participant	After the user has captured the details in FR301, the system should randomly assign the participant to one of two experiment groups (SMS or control)	NA	N.A.	NA	TC302				
-------	-----------------------	--	----	------	----	-------	--	---	---	---

FR303	Send 'Add Recipient to Campaign' Request to Mobilisr	If the participant was randomised to the SMS group, the system must send a 'Add Recipient to Campaign' request to Mobilisr with the following details (Campaign Name / ID, Mobile Number, First Name, Surname, Mobile Organisation, Mobilisr Username, Mobilisr Password). Mobilisr should then send a response to this request ('OK', 'Queued' or 'Error').	NA	N.A.	NA	TC303				
-------	--	--	----	------	----	-------	--	--	--	--

## **Remove Participant from Study (FR400)**

Req No	Req Name	Requirement Description (includes Business Rules)	Process Diagram	GUI Mockup	Use Case	Test Case		Design (BA)	Dev (Dev)	Test (BA/Dev)
--------	----------	--	-----------------	------------	----------	-----------	--	-------------	-----------	---------------

FR401	Remove Participant from Study	A user with study permission must be able to remove a participant from the study. The participant may be in either group of the study. Set the stop date in the studyParticipant table - don't delete it.	NA	GM401	UC401	TC401				
-------	-------------------------------	---	----	-------	-------	-------	--	--	--	--

FR402	Send 'Remove Recipient from Campaign' Request to Mobilisr	If the participant to be removed (in FR401) is in the SMS group, the system must send a 'Remove Recipient from Campaign' request to Mobilisr with the following details (Campaign Name / ID, Mobile Number, First Name, Surname, Mobile Organisation, Mobilisr Username, Mobilisr Password). Mobilisr should then send a response to this request ('OK', 'Queued' or 'Error').	NA	NA	NA	TC402				
-------	---	--	----	----	----	-------	--	--	--	--

#### **Reminder Module (FR500)**

Req No	Req Name	Requirement Description (includes Business Rules)	Process Diagram	GUI Mockup	Use Case	Test Case		Design (BA)	Dev (Dev)	Test (BA/Dev)
FR501	Create Property File for Appointment Reminder Message Templates	The messages that will be sent out for scheduled appointment reminders (e.g. 2 weeks before pickup, 1 week before, 1 day after, etc.) will need to be saved in iDART and relatively accessible. We do not require a screen for this, but the more high-level user should be able to modify these message templates as well as the number of days before and after drug collections.	NA	NA	NA	NA			RL	

FR502	Flag participants that are to receive reminder messages	The system must automatically flag all participating patients on days prior to their appointment date. The days are specified in the SMS properties file. The flagged patients indicate which patients are to receive a reminder message on that day.	NA	NA	NA	NA			RL	
FR503	Flag late patients	The system must automatically flag all participants who have missed their appointment by "x" amount of days. The days late is specified in the SMS properties file.	NA	NA	NA	NA			RL	
FR504		Allow the user to send a 'Just send	NA	NA	NA	TC504				

Send 'Just Send SMS' Request to Mobilisr (with generate d lists)	SMS' Request to Mobilisr with the list of patients flagged. The following information will be required: Mobilisr Organisation, Mobilisr Username, Mobilisr Password , Campaign Name, Time to send, Recipient List (this will include Mobile Number per Participant, First Name, Surname) . Mobilisr should then send a response to this request ( <b>not sure what to do with this response however since this is automatic and not reliant on the user -</b>				

		see <b>ALERTS</b> table)							
--	--	--------------------------------	--	--	--	--	--	--	--

## Appointments (FR600)

Req No	Req Name	Requirement Description (includes Business Rules)	Process Diagram	GUI Mockup	Use Case	Test Case		Design (BA)	Dev (Dev)	Test (BA/Dev)
FR601	Issue an appointment date for a new patient	Allow the user to create a new patient and issuing an appointment date for the patient API Link	NA	NA	UC601	TC601				
FR602	Update an existing patient's appointment date	Allow the user to update an existing patient's appointment date in the update patient screen API Link	NA	NA	UC602	TC602				
FR603	Dispense directly to a patient who has been on an appointment	Allow the user to dispense directly to a patient to ensure that the correct appointment date as well as the correct visit date has been created API Link	NA	NA	UC603	TC603				

FR604	Dispense to an existing patient for a later pickup	Allow the user to dispense to an existing patient for a later pickup to ensure that no new appointments are created i.e. appointment date must not change API Link	NA	NA	UC604	TC604			
FR605	Deletion of appointments								

## **Study Alerts (FR700)**

Req No	Req Name	Requirement Description (includes Business Rules)	Process Diagram	GUI Mockup	Use Case	Test Case		Design (BA)	Dev (Dev)	Test (BA/Dev)
FR701	Study Alerts	Show the user all alerts that have occurred, as stored in the alerts database table.	N.A.	GM701						

## Reports (FR800)

Req No	Req Name	Requirement Description (includes Business Rules)	Process Diagram	GUI Mockup	Use Case	Test Case		Design (BA)	Dev (Dev)	Test (BA/Dev)
--------	----------	--	-----------------	------------	----------	-----------	--	-------------	-----------	---------------

FR801	Schedule d Appointm ents and Attenda nce Report	On JasperRe ports, the following report needs to be accessible.	N.A.	GM801						
-------	---	---	------	-------	--	--	--	--	--	--

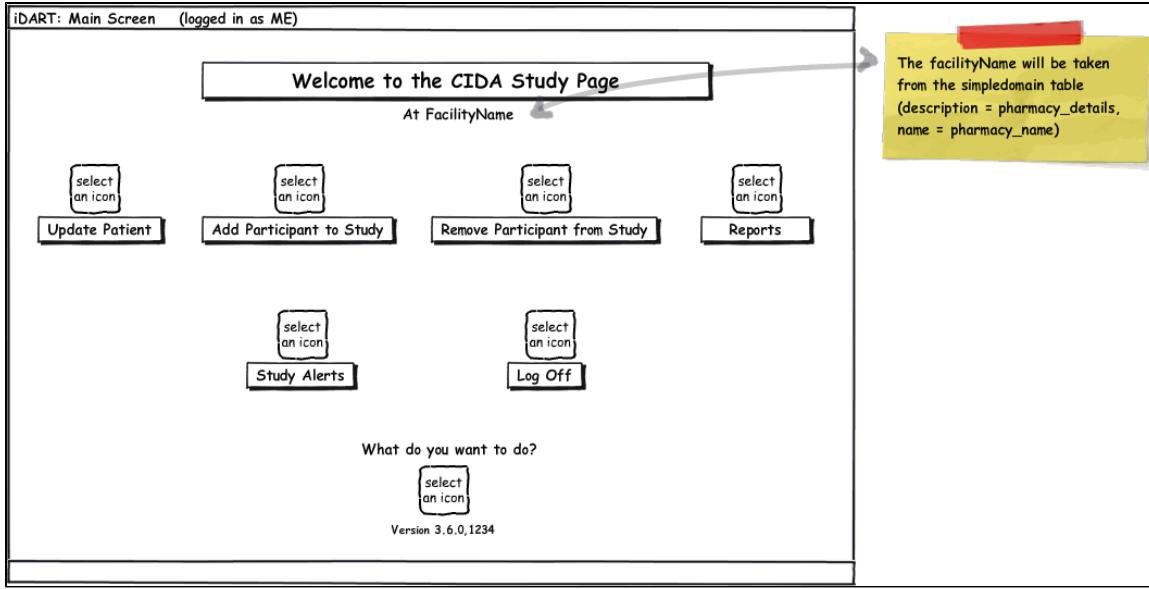
## Parking Lot (including notes for EMIT & Mobilisr requirements)

	Mobilisr Reports	<ul style="list-style-type: none"> <li>• How many patients have been added or removed from a campaign</li> <li>• Just send SMS - list of all participants? including first name, surname, etc.</li> </ul>								
		What to do when people change their numbers? ?? From both iDART and Mobilisr side of things???								

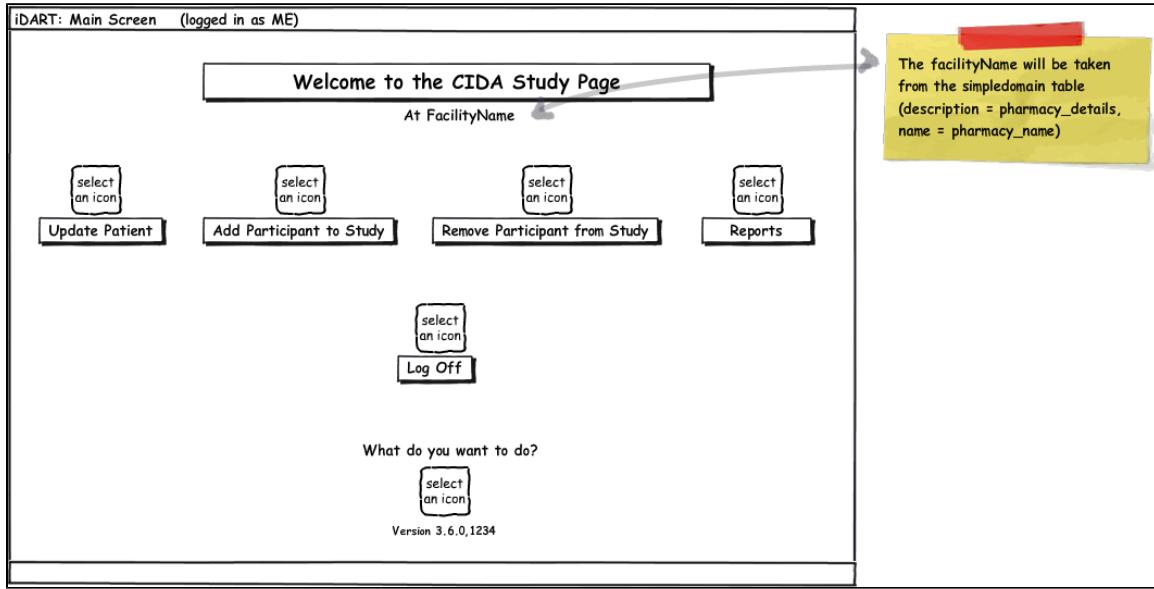
## CIDA GUI Mockups

### GM102 Welcome Page for StudyWorker User

Updated GUI: Based on team discussions to incorporate the 'alerts table' 09 May 2011



Original GUI:



## GM103 Welcome Page for ReportsWorker

iDART: Main Screen (logged in as ME)

Welcome to the Reports Page  
At FacilityName

What do you want to do?

Version 3.6.0,1234

The facilityName will be taken from the simpledomain table  
(description = pharmacy\_details, name = pharmacy\_name)

## GM104 Add New User (CIDA)

iDART: Add a New User (logged in as ME)

Add a new iDART User  
All fields marked with \* are compulsory

Add New iDART User  Update Current User

* Username: <input type="text"/>	Configure Clinic Access: <input checked="" type="checkbox"/> Clinic A <input type="checkbox"/> Clinic B <input type="checkbox"/> Clinic C <input type="checkbox"/> Clinic D
* Password: <input type="password"/>	
* Confirm Password: <input type="password"/>	
Note this is for study purposes only. Leave blank for pharmacy staff	
Configure User Type: <input type="radio"/> study worker <input type="radio"/> access to reports	
<input type="button" value="Save"/> <input type="button" value="Clear"/> <input type="button" value="Cancel"/>	

This group will only be shown if the property CIDA = true

The facilityName will be taken from the simpledomain table  
(description = pharmacy\_details, name = pharmacy\_name)

When the page loads, neither radio button should be selected

Pressing clear must also remove any selected radio button in the 'Configure User Type' group

## GM301 Add Participant to Study

Updated 02 June 2011 (Based on specifying language for appointment reminders)

**Add Participant to Study**

These fields will NOT be editable

Patient ID: [ ] Patient Search: [ ]

First Names: [ ] Use this format, 07 Mar 1980 (31 yrs)

Surname: [ ]

Date of Birth & Age: [ ] Use this format, 07 Feb 2011

Sex: [ ]

ARV Initiation Date: [ ]

First Drug Collection: [ ]

All fields marked with \* are compulsory

- \* Cellphone Number: [ ]
- \* Network: [please select ...]
- \* Alternate Cellphone Number: [ ]
- \* Informed Consent Given:  Yes
- \* Language for Appointment Reminders: [please select ...]
- \* Available Campaigns: [please select ...]

[refresh list of available campaigns](#)

**PLEASE REMEMBER TO COMPLETE THE BASELINE QUESTIONNAIRE ONLINE**

**Add to Study** **Clear** **Cancel**

After pressing this button, the following checks will be done to ensure eligibility of study:
 

- \* participant is above age of 18
- \* new to ARVs (ART naïve)
- \* informed consent has been given
- \* participant isn't already enrolled in the study

If successful, a randomisation algorithm (FR302) will be applied and the participant will be randomly selected for the experimental or control arm of the study.

If the participant is selected for the experimental arm, send a 'Add Recipient to Campaign' Request (FR303) to Mobilis with this participant's details.

The standard iDART patient search applies. All patients should be shown in the search window.

Male, Female or Unknown

3TC 150, EFV 600, D4T30 collected at 2:30pm on Mon, 20 Feb 2011

Select the network for the cellphone number - Vodacom : MTN : Cell C : Virgin : 8ta

Cellphone validity checks (correct format, only numbers, etc.)

The language for appointment reminders will be read from the 'sms.properties' file. For this study, only English and isiZulu will be required.

The system will list the campaigns available. English and isiZulu versions will be available (e.g. Charles James isiZulu, Charles James English). For this study, only one campaign will be selected per participant. When this page loads, check that the list is populated. If it isn't, send a request to Mobilis to get the list of available campaigns (FR201) & populate the drop down list accordingly.

Pressing this link will send a request to Mobilis to get a list of available campaigns (FR201) & populate the drop down list accordingly.

**Add Participant to Study**

The standard iDART patient search applies. All patients should be shown in the search window.

These fields will NOT be editable

Patient ID: [ ] Patient Search: [ ]

First Names: [ ] Use this format, 07 Mar 1980 (31 yrs)

Surname: [ ]

Date of Birth & Age: [ ] Use this format, 07 Feb 2011

Sex: [ ]

ARV Initiation Date: [ ]

First Drug Collection: [ ]

All fields marked with \* are compulsory

- \* Cellphone Number: [ ]
- \* Network: [please select ...]
- \* Alternate Cellphone Number: [ ]
- \* Informed Consent Given:  Yes
- \* Available Campaigns: [please select ...]

[refresh list of available campaigns](#)

**PLEASE REMEMBER TO COMPLETE THE BASELINE QUESTIONNAIRE ONLINE**

**Add to Study** **Clear** **Cancel**

After pressing this button, the following checks will be done to ensure eligibility of study:
 

- \* participant is above age of 18
- \* new to ARVs (ART naïve)
- \* informed consent has been given
- \* participant isn't already enrolled in the study

If successful, a randomisation algorithm (FR302) will be applied and the participant will be randomly selected for the experimental or control arm of the study.

If the participant is selected for the experimental arm, send a 'Add Recipient to Campaign' Request (FR303) to Mobilis with this participant's details.

Select the network for the applicable cellphone number, the user selects the network from the drop down list which contains the following networks. Vodacom : MTN : Cell C : Virgin : 8ta

Cellphone validity checks (correct format, only numbers, etc.)

The system will list the campaigns available. English and isiZulu versions will be available (e.g. Charles James isiZulu, Charles James English). For this study, only one campaign will be selected per participant. When this page loads, check that the list is populated. If it isn't, send a request to Mobilis to get the list of available campaigns (FR201) & populate the drop down list accordingly.

Pressing this link will send a request to Mobilis to get a list of available campaigns (FR201) & populate the drop down list accordingly.

Updated 04 May 2011

Updated based on discussions in SPRINT planning (01 Mar 2011)

**Add Participant to Study**

Patient ID	<input type="text"/>	Patient Search
First Names	<input type="text"/>	
Surname	<input type="text"/>	
Date of Birth & Age	<input type="text"/>	
Sex	<input type="text"/>	
ARV Initiation Date	<input type="text"/>	
First Drug Collection	<input type="text"/>	
All fields marked with * are compulsory		
* Cellphone Number	<input type="text"/>	
* Network	please select ... <input type="button"/>	
* Alternate Cellphone Number	<input type="text"/>	
* Informed Consent Given	<input type="checkbox"/> Yes	<input type="text"/>
* Available Campaigns	please select ... <input type="button"/>	
refresh list of available campaigns		
PLEASE REMEMBER TO COMPLETE THE BASELINE QUESTIONNAIRE ONLINE		
<input type="button"/> Add to Study <input type="button"/> Clear <input type="button"/> Cancel		

After pressing this button, the following checks will be done to ensure eligibility of study:  
 \* participant is above age of 18  
 \* new to ARVs (ART naïve)  
 \* informed consent has been given  
 \* participant isn't already enrolled in the study

If successful, a randomisation algorithm (FR302) will be applied and the participant will be randomly selected for the experimental or control arm of the study.

If the participant is selected for the experimental arm, send a 'Add Recipient to Campaign' Request (FR303) to Mobilisr with this participant's details.

The standard iDART patient search applies. All patients should be shown in the search window.

These fields will NOT be editable

Use this format, 07 Mar 1980 (31 yrs)

Male, Female or Unknown

Use this format, 07 Feb 2011

3TC 150, EFV 600, D4T30 collected at 2:30pm on Mon, 20 Feb 2011

Cellphone validity checks (correct format, only numbers, etc.)

Select the network for the applicable cellphone number. The user selects the network from the drop down list which contains the following networks: Vodacom : MTN : Cell C : Virgin : Bta

The system will list the campaigns available. English and isiZulu versions will be available (e.g. Charles James isiZulu, Charles James English). For this study, only one campaign will be selected per participant. When this page loads, check that the list is populated. If it isn't, send a request to Mobilisr to get the list of available campaigns (FR201) & populate the drop down list accordingly.

Pressing this link will send a request to Mobilisr to get a list of available campaigns (FR201) & populate the drop down list accordingly.

## Original GUI

**Add Participant to Study**

Patient ID	<input type="text"/>	Patient Search
First Names	<input type="text"/>	
Surname	<input type="text"/>	
Date of Birth & Age	<input type="text"/>	
Sex	<input type="text"/>	
ARV Initiation Date	<input type="text"/>	
First Drug Collection	<input type="text"/>	
All fields marked with * are compulsory		
* Cellphone Number	<input type="text"/>	
* Informed Consent Given	<input type="checkbox"/> Yes	<input type="text"/>
* Available Campaigns	please select ... <input type="button"/>	
refresh list of available campaigns		
<input type="button"/> Add to Study <input type="button"/> Clear <input type="button"/> Cancel		

After pressing this button, the following checks will be done to ensure eligibility of study:  
 \* participant is above age of 18  
 \* new to ARVs (ART naïve)  
 \* informed consent has been given  
 \* participant isn't already enrolled in the study

If successful, a randomisation algorithm (FR302) will be applied and the participant will be randomly selected for the experimental or control arm of the study.

If the participant is selected for the experimental arm, send a 'Add Recipient to Campaign' Request (FR303) to Mobilisr with this participant's details.

The standard iDART patient search applies. All patients should be shown in the search window.

These fields will NOT be editable

Use this format, 07 Mar 1980 (31 yrs)

Male, Female or Unknown

Use this format, 07 Feb 2011

3TC 150, EFV 600, D4T30 collected at 2:30pm on Mon, 20 Feb 2011

Cellphone validity checks (correct format, only numbers, etc.)

The system will list the campaigns available. English and isiZulu versions will be available (e.g. Charles James isiZulu, Charles James English). For this study, only one campaign will be selected per participant. When this page loads, check that the list is populated. If it isn't, send a request to Mobilisr to get the list of available campaigns (FR201) & populate the drop down list accordingly.

Pressing this link will send a request to Mobilisr to get a list of available campaigns (FR201) & populate the drop down list accordingly.

## GM401 Remove Participant from Study

**Remove Participant from Study**

Patient ID	<input type="text"/>	Patient Search
First Names	<input type="text"/>	
Surname	<input type="text"/>	
Date of Birth & Age	<input type="text"/>	
Sex	<input type="text"/>	
ARV Initiation Date	<input type="text"/>	
First Drug Collection	<input type="text"/>	

All fields marked with \* are compulsory

\* Cellphone Number

\* Study Enrolment Date

PLEASE REMEMBER TO COMPLETE THE OPT-OUT INTERVIEW ONLINE

Note that the participant can be from either study arm.  
 \* If the study arm = control, capture their opt-out date.  
 \* If the study arm = experiment (i.e. SMS), capture their opt-out date and send a 'Remove Recipient from Campaign' Request to Mobilisr (FR402) to Mobilisr with this participant's details.

The search window should be filtered to show only study participants. This would include participants in both arms of the study.

These fields will NOT be editable

Use this format, 07 Mar 1980 (31 yrs)

Male, Female or Unknown

Use this format, 07 Feb 2011

3TC 150, EFV 600, D4T30 collected at 2:30pm on Mon, 20 Feb 2011

Use this format, 07 Feb 2011

## GM701 Study Alerts

**Study Alerts**

Date	Alert Message
09 May 2011 at 10:30am	Failed to add participant 123 (Sarah Brown) to study
11 May 2011 at 07:00am	Failed to send appointment reminders to participants

Order the table by most recent activity at the top of the table. The columns should be sortable (via click) if possible.

Show as much information as possible

Pressing the 'Clear' button will remove the alerts from the alerts database table. The user needs to confirm they want to make this delete - see the confirmation dialog box below.

**Delete All Alerts in the Table?**

By clearing this table, you will be deleting the alerts from the table in the database. This action cannot be undone, and you should only remove these alerts if you have permission to do so, from a Cell-Life staff member. Are you sure you want to delete these alerts?

Clicking 'Yes' will delete all the alerts from the database alerts table. The user will be taken back to the main welcome screen of the study worker.

If the user clicks 'No', the database table will not be affected. The user will be taken back to the Alerts screen above.

## GM801 Scheduled Appointments and Attendance Report

The figure consists of three screenshots of a software application interface, each with associated yellow callout boxes containing handwritten notes.

- Scheduled Appointments and Attendance Report:** Shows fields for "Scheduled Appointments from" and "to" with date pickers, and buttons for "Generate Report" and "Cancel". A note says: "The dates should read appointments that were scheduled in this time, and not look at the dates of actual attendance. The actual attendance might be after this date and should be included in the tables below if that data is available."
- Scheduled Appointments and Attendance Report for pharmacyName:** Shows a table of appointment counts for various lateness categories and a total. A note says: "The report is broken up into two identical tables. It is separated based on whether participants are in the CIDA study or not. This can be seen in the studyParticipant database table." Another note specifies: "PharmacyName is taken from the iDART database" and "Use date format of 04 Jun 2011 for startDate and endDate in this title".
- Control Group:** Shows a table of appointment counts for various lateness categories and a total. A note says: "For rows 2 and after, the number of days late should be determined by the actual database, not these hard-coded numbers displayed here. For example, if one patient was 26 days late, they should be shown in a row called 'Appointments in which patients were 26 days later'. That way, the report won't necessarily have numbers 1,2,3,4,5,6,7."

## CIDA Test Cases

### TC104 Add a new user (roles)

Check that the administrator can add roles to user accounts

#### Prerequisites

1. The Administrator must be logged into the system

#### Test Data

Username : JohnS

Password : 123

Role : Study Worker / Report Worker

#### Test Matrix

Test Scenario	Test steps	Outcome	Comments (include link to Jira ticket if failure)
---------------	------------	---------	---

TC104A - create user account (Study Worker)	<ul style="list-style-type: none"> <li>• enter test data</li> <li>• Select one of the roles only</li> <li>• Save</li> <li>• Logoff* Login with new user account</li> <li>• Check for corresponding Welcome screen</li> </ul>	pass	
TC104B - create user account (Report Worker)	<ul style="list-style-type: none"> <li>• enter test data</li> <li>• Select one of the roles only</li> <li>• Save</li> <li>• Logoff* Login with new user account</li> <li>• Check for corresponding Welcome screen</li> </ul>	pass	
TC104C - Create User Account (Pharmacist)	<ul style="list-style-type: none"> <li>• enter test data</li> <li>• Leave roles blank</li> <li>• Save</li> <li>• Logoff* Login with new user account</li> <li>• Confirm that user can view all functions</li> </ul>	pass	
TC104D - Confirm Role selection	<ul style="list-style-type: none"> <li>• Confirm that the role section has no options selected by default</li> </ul>	pass	
TC104E - Clear fields	<ul style="list-style-type: none"> <li>• Enter fields in all sections</li> <li>• Click the Clear button</li> <li>• Confirm that all fields have been cleared including the role selection.</li> </ul>	pass	

## TC201 Send Request for Campaign list

Use Case Number	TC201
Use Case Name	Request for Campaign list
Goal in Context	All active campaigns created within Mobilisr is displayed in the drop downlist in iDART
Scope	
Level	
Precondition	<ol style="list-style-type: none"> <li>1. CIDA study worker has admin user account in Mobilisr</li> <li>2. Study worker login created in iDART</li> </ol>
Success End Condition	study worker is able to view all the active campaigns in iDART
Failed End Condition	campaigns are not displayed
Primary Actor	study worker
Trigger	<ol style="list-style-type: none"> <li>1. Study worker clicks the refresh link in iDART</li> </ol>
Input	

<b>Main Success Scenario</b>	<ol style="list-style-type: none"> <li>1. Log into Mobilisr</li> <li>2. Click on the admin section</li> <li>3. click on Campaigns</li> <li>4. Add new campaign</li> <li>5. Start newly created campaign</li> <li>6. log into iDART as study worker</li> <li>7. Click add participant to study</li> <li>8. Click the "refresh list of available campaigns" link</li> <li>9. Click the drop down list to confirm that the newly created campaign is available.</li> </ol>
<b>Subvariations (Alternate Course of Action)</b>	
<b>Extensions</b>	N/A
<b>Superordinate Use Cases</b>	UC101 ; UC104;
<b>Subordinate Use Cases</b>	N/A
<b>Secondary Actors</b>	N/A
<b>Open Issues</b>	N/A

## TC301 Add a Participant to Study

Check that a patient's appointment date can be updated in iDART

### Prerequisites

1. The study worker user must be logged into the system
2. The patient must exist in the system
3. Campaign must exist in Mobilisr

### Test Matrix

Test Scenario	Test Data	Outcome	Comments (include link to Jira ticket if failure)
TC301A - Adding a participant to a study	Perform UC301	Success	
TC301B - Consent not given	Leave "Informed Consent Given" unchecked	Success	
TC301C - participant already added	Choose a patient that has already been added. Should be unable to be added.	Success	
TC301D - Invalid mobile number	Change the mobile number of the patient to an invalid format. Add button should be greyed out	Success	
TC301E - participant is only added to one campaign	A participant is cannot be added to another campaign in the list if already added a CIDA campaign.		

## FR301 Add Participant to Study

### Business Rules & Error Messages

Business Rule	Business Rule Description	Error Message		
BR301A		A patient cannot be added to study if they already added		The patient you are trying to add to a study, has already been added to a different study.
BR301B		The mobile number entered in iDART must be in the format accepted by Mobilisr. No special characters and no spaces included in the numbert		The mobile number of this participant is in the wrong format. Please remove all spaces and special characters. eg 0821234567
BR301C		A participant cannot be added to another campaign in the list if already added a CIDA campaign. A Participant can only partake in one CIDA campaign at a time.		The patient you are trying to add to a study, has already been added to a different study.

## TC302 Randomise participants

Use Case Number	TC302
Use Case Name	Randomise Participant
Goal in Context	A patient that is added to the study is randomly determined whether they will be in control group or not.
Scope	
Level	
Precondition	1. Patient's need to be created
Success End Condition	participant is randomly entered into the study or control group
Failed End Condition	no randomisation is done
Primary Actor	study worker
Trigger	1. Patient signs study consent form
Input	

<b>Main Success Scenario</b>	<ol style="list-style-type: none"> <li>1. Log into iDART</li> <li>2. Click on 'Add Participant From Study' Button</li> <li>3. Select a male patient from the list</li> <li>4. Click the "Add to study" button</li> <li>5. Repeat so 3 and 4 so that 10 males have been added</li> <li>6. Do the same for 10 females.</li> <li>7. log into the database and view the StudyParticipant table to confirm that there are 10 males in the control group and 10 males in the experiment group. The same goes for the females</li> </ol>
<b>Subvariations (Alternate Course of Action)</b>	3-4 If the user does not want to dispense to the patient, he can click on the 'Cancel' button to cancel the operation at each of these steps.
<b>Extensions</b>	N/A
<b>Superordinate Use Cases</b>	UC101 ; UC104;
<b>Subordinate Use Cases</b>	N/A
<b>Secondary Actors</b>	N/A
<b>Open Issues</b>	N/A

## TC303 Send 'Add Recipient to Campaign' Request to Mobilisr

<b>Use Case Number</b>	TC303
<b>Use Case Name</b>	Send 'Add Recipient to Campaign' Request to Mobilisr
<b>Goal in Context</b>	All active campaigns created within Mobilisr is displayed in the drop downlist in iDART
<b>Scope</b>	
<b>Level</b>	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CIDA study worker has admin user account in Mobilisr</li> <li>2. Study worker login created in iDART</li> </ol>
<b>Success End Condition</b>	study worker is able to view the recipient that was added in iDART
<b>Failed End Condition</b>	recipient not visible in the campaign
<b>Primary Actor</b>	study worker
<b>Trigger</b>	<ol style="list-style-type: none"> <li>1. Study worker clicks the "Add participant to study" button in iDART</li> </ol>
<b>Input</b>	Patient from iDART

<b>Main Success Scenario</b>	<ol style="list-style-type: none"> <li>1. Log into iDART as study worker</li> <li>2. Click on the "Add Patient to Study"</li> <li>3. Scan the patientID / search and select the patient from the "Patient Search"</li> <li>4. Check the "informed consent given" checkbox</li> <li>5. Select campaign from drop down list</li> <li>6. Confirm message schedule/ Select message schedule from dropdown list</li> <li>7. Click add participant to study</li> <li>8. Log into Mobilisr using the CIDA admin login</li> <li>9. Navigate to the campaigns screen.</li> <li>10. Select the campaign to which the patient was added</li> <li>11. Click on the "manage recipients" button/icon</li> <li>12. Confirm that the patients contact number has been added to the campaign</li> </ol>
<b>Subvariations (Alternate Course of Action)</b>	
<b>Extensions</b>	N/A
<b>Superordinate Use Cases</b>	UC101 ; UC104;
<b>Subordinate Use Cases</b>	N/A
<b>Secondary Actors</b>	N/A
<b>Open Issues</b>	N/A

## TC401 Remove Participant from Study

Check that a patient's appointment date can be updated in iDART

### Prerequisites

1. The study worker user must be logged into the system
2. The patient must exist in the system
3. Campaign must exist in Mobilisr
4. Patient must have already been added to the campaign

### Test Matrix

Test Scenario	Test Data	Outcome	Comments (include link to Jira ticket if failure)
TC401A - Remove participant from a study	Perform UC401	failed	<a href="http://jira.cell-life.org/browse/IDA-RT-230">http://jira.cell-life.org/browse/IDA-RT-230</a>
TC401B - Clear screen	Select the participant and then test the Clear button to remove everything that was added.	Success	
TC401C - patient not on study	attempt to remove a patient that is not on any CIDA study. Should not be allowed	Success	

## FR401 Remove Participant from Study

### Business Rules & Error Messages

Business Rule	Business Rule Description	Error Message				
BR401A	Only patients that are on an active CIDA study can be removed	The patient that you are trying to remove is currently not on any study	.			

## TC402 Send 'Remove Recipient from Campaign' Request to Mobilisr

Use Case Number	TC402
Use Case Name	Send 'Remove Recipient from Campaign' Request to Mobilisr
Goal in Context	Patients removed from stud in iDART are removed from the campaign in Mobilisr
Scope	
Level	
Precondition	<ul style="list-style-type: none"> <li>1. CIDA study worker has admin user account in Mobilisr</li> <li>2. Study worker login created in iDART</li> <li>3. Patient has been added to a study</li> </ul>
Success End Condition	Patients has been removed from the campaign
Failed End Condition	recipient still in the campaign
Primary Actor	study worker
Trigger	<ul style="list-style-type: none"> <li>1. Study worker clicks the "Remove participant from study" button in iDART</li> </ul>
Input	Patient from iDART
Main Success Scenario	<ul style="list-style-type: none"> <li>1. Log into iDART as study worker</li> <li>2. Click on the "Remove Participant from Study"</li> <li>3. Scan the patientID / search and select the patient from the "Patient Search"</li> <li>4. Check the "informed consent given" checkbox</li> <li>5. Select campaign from drop down list</li> <li>6. Confirm message schedule/ Select message schedule from dropdown list</li> <li>7. Click "Remove from Study" button</li> <li>8. Log into Mobilisr using the CIDA admin login</li> <li>9. Navigate to the campaigns screen.</li> <li>10. Select the campaign to which the patient was added</li> <li>11. Click on the "manage recipients" button/icon</li> <li>12. Confirm that the patient's contact number has been removed from the campaign</li> </ul>
Subvariations (Alternate Course of Action)	

<b>Extensions</b>	N/A
<b>Superordinate Use Cases</b>	UC101 ; UC104;
<b>Subordinate Use Cases</b>	N/A
<b>Secondary Actors</b>	N/A
<b>Open Issues</b>	N/A

## TC504 Send 'Just Send SMS' Request to Mobilisr

<b>Use Case Number</b>	TC504
<b>Use Case Name</b>	Send 'Just Send SMS' Request to Mobilisr
<b>Goal in Context</b>	Create a "Just SMS Campaign" to remind participants of their appointment date.
<b>Scope</b>	
<b>Level</b>	
<b>Precondition</b>	<ul style="list-style-type: none"> <li>1. CIDA study worker has admin user account in Mobilisr</li> <li>2. Study worker login created in iDART</li> <li>3. A patient created on Mobilisr with a valid prescription has been added to the study.</li> </ul>
<b>Success End Condition</b>	A campaign has been created in Mobilisr with the participant added to that campaign
<b>Failed End Condition</b>	campaign not created
<b>Primary Actor</b>	study worker
<b>Trigger</b>	<ul style="list-style-type: none"> <li>1. Participant appointment is due in x days</li> <li>2. participant has missed their appointment for x days</li> </ul>
<b>Input</b>	Patient appointment date

<b>Main Success Scenario</b>	<ol style="list-style-type: none"> <li>1. Log into iDART as study worker</li> <li>2. Click on the "Add Patient to Study"</li> <li>3. Scan the patientID / search and select the patient from the "Patient Search"</li> <li>4. Check the "informed consent given" checkbox</li> <li>5. Select campaign from drop down list</li> <li>6. Confirm message schedule/ Select message schedule from dropdown list</li> <li>7. Click add participant to study</li> <li>8. Log into Mobilisr using the CIDA admin login</li> <li>9. Navigate to the campaigns screen.</li> <li>10. Select the campaign to which the patient was added</li> <li>11. Click on the "manage recipients" button/icon</li> <li>12. Confirm that the patients contact number has been added to the campaign</li> <li>13. log into iDART as the pharmacist</li> <li>14. navigate to the "Make up drug packages for patients Screen"</li> <li>15. Create a package for the participant on the study</li> <li>16. Change the next appointment date so that the date will coincide with the value indicated in the sms.properties file</li> <li>17. Dispense the package</li> <li>18. Within Mobilisr Navigate to the "Just Send SMS" screen and confirm that the campaign has been created and the user has been added to that campaign.</li> </ol>
<b>Subvariations (Alternate Course of Action)</b>	
<b>Extensions</b>	N/A
<b>Superordinate Use Cases</b>	UC101 ; UC104;
<b>Subordinate Use Cases</b>	N/A
<b>Secondary Actors</b>	N/A
<b>Open Issues</b>	N/A

## TC601 - Issue an appointment date for a new patient

Check that a patient can be created and an appointment date can be issued in iDART

### Prerequisites

1. The user must be logged into the system

### Test Matrix

Test Scenario	Test Data	Outcome	Comments (include link to Jira ticket if failure)
TC601A - Adding a patient	Enter patient's details	Success	
TC601B - Not entering First Name	Enter patient's details, omitting First Name	Success	
TC601C - Dispense medication to patient	Enter drug to be dispensed	Success	
TC601E - Entering an appointment date in the future	Enter a date greater than today's date	Success	
TC601F - Entering an appointment date in the past	Enter a date that is less than today's date	Failed	

**Reference to Business Rules (FR 601 - Issue an appointment date for a new patient )**

FR 601 - Issue an appointment date for a new patient		
Business Rules & Error Messages		
Business Rule	Business Rule Description	Error Message
BR601A	Patient must have been created	
BR601B	Patient must have had medication dispensed to him	
BR601C	Appointment date must be in the future. If the date is in the past the following error message should be displayed.	You can't set an appointment date in the past. Please select a date after today.

## TC602 - Update an existing patient's appointment date

Check that a patient's appointment date can be updated in iDART

### Prerequisites

1. The user must be logged into the system
2. The patient must exist in the system

### Test Matrix

Test Scenario	Test Data	Outcome	Comments (include link to Jira ticket if failure)
TC601A - Adding an appointment without selecting the patient	Click on the 'Treatment Management' tab	Success	
TC601B - Dispense medication to patient	Enter drug to be dispensed	Success	
TC601C - Entering an appointment date in the future	Enter a date greater than today's date	Success	
TC601D - Entering an appointment date in the past	Enter a date that is less than today's date	Failed	

**Reference to Business Rules ([IDART:FR 602 - Update an existing patient's appointment date](#))**

The page FR 602 - Update an existing patient's appointment date could not be found.

## TC603 - Dispense directly to a patient who has been on an appointment

Check that the user can dispense directly to a patient and ensure that the correct appointment date as well as the correct visit date has been created

### Prerequisites

1. The User must be logged in to the system.
2. The patient must exist in the system
3. The patient must have been on an appointment

### Test Matrix

Test Scenario	Test Data	Outcome	Comments (include link to Jira ticket if failure)

TC601A - Select Patient	Enter patient's ID in the patient Search	Success	
TC601B - Dispense medication to patient	Enter drug to be dispensed	Success	
TC601E - Entering an appointment date in the future	Enter a date greater than today's date	Success	
TC601F - Entering an appointment date in the past	Enter a date that is less than today's date	Failed	

Reference to Business Rules (FR 603 - Dispense directly to a patient who has been on an appointment )

FR 603 - Dispense directly to a patient who has been on an appointment					
Business Rules & Error Messages					
Business Rule	Business Rule Description	Error Message			
BR603A	Patient must exist				
BR603B	Patient must have had medication dispensed to him				
BR603C	Appointment date must be in the future. If the date is in the past the following error message should be displayed	Please select a date greater than today's date.			
BR603D	Visit date must be equal to the date the patient visited the hospital				

## TC604 - Dispense to an existing patient for a later pickup

Check that the user can dispense to an existing patient for a later pickup to ensure that no new appointments are created i.e. appointment date must not change

### Prerequisites

1. The User must be logged in to the system.
2. The patient must exist in the system

### Test Matrix

Test Scenario	Test Data	Outcome	Comments (include link to Jira ticket if failure)
---------------	-----------	---------	---

TC601A - Select Patient	Enter patient's ID in the patient Search	Success	
TC601B - Create drug package	Enter drug to be dispensed	Success	
TC601C - Entering an appointment date	Click on 'Next Appointment Button'	Success	Could not add a new appointment.

#### Reference to Business Rules ([FR604 - Dispense to an existing patient for a later pickup](#) )

The page FR604 - Dispense to an existing patient for a later pickup could not be found.

## CIDA Use Cases

### UC101 Study worker Welcome page

Use Case Number	UC101
Use Case Name	Study Worker Welcome Page
Goal in Context	Change the welcome screen based on the user role assigned to the user.
Scope	
Level	
Preconditions	<ul style="list-style-type: none"> <li>User needs to be created and be assigned a role of Study worker</li> </ul>
Success End Condition	User is logged in to the system and sees the required welcome screen for study workers
Failed End Condition	User logs in and sees a welcome screen for a role other than a study worker
Primary Actor	study worker user
Trigger	login
MAIN SUCCESS SCENARIO	<ol style="list-style-type: none"> <li>Start the iDART application</li> <li>Enter username</li> <li>Enter password</li> <li>Click 'Login' button</li> </ol>
EXTENSIONS	
SUB-VARIATIONS	
Superordinate Use Cases	Login
Subordinate Use Cases	
Secondary Actors	
OPEN ISSUES	<ol style="list-style-type: none"> <li>Only one instance of a user can be logged in at one point in time. That is, the system should not allow the same user details to be used at the same time.</li> </ol>

### UC102 Welcome screen for Reports Worker

Use Case Number	UC102
Use Case Name	Study Worker Welcome Page

<b>Goal in Context</b>	Change the welcome screen based on the user role assigned to the user.
<b>Scope</b>	
<b>Level</b>	
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>• User needs to be created and be assigned a role of Reports worker</li> </ul>
<b>Success End Condition</b>	User is logged in to the system and sees the required welcome screen for reports workers
<b>Failed End Condition</b>	User logs in and sees a welcome screen for a role other than a reports worker
<b>Primary Actor</b>	report worker user
<b>Trigger</b>	login
<b>MAIN SUCCESS SCENARIO</b>	<ol style="list-style-type: none"> <li>1. Start the iDART application</li> <li>2. Enter username</li> <li>3. Enter password</li> <li>4. Click 'Login' button</li> </ol>
<b>EXTENSIONS</b>	
<b>SUB-VARIATIONS</b>	
<b>Superordinate Use Cases</b>	Login
<b>Subordinate Use Cases</b>	
<b>Secondary Actors</b>	
<b>OPEN ISSUES</b>	<ol style="list-style-type: none"> <li>1. Only one instance of a user can be logged in at one point in time. That is, the system should not allow the same user details to be used at the same time.</li> </ol>

## UC104 Add new user

<b>Use Case Number</b>	UC104
<b>Use Case Name</b>	Add new user
<b>Goal in Context</b>	Create users that have different roles
<b>Scope</b>	
<b>Level</b>	
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>• iDART application needs to be installed</li> </ul>
<b>Success End Condition</b>	User is able to assign roles to user accounts
<b>Failed End Condition</b>	User unable to assign roles
<b>Primary Actor</b>	admin user
<b>Trigger</b>	report worker / study worker needs access to iDART

<b>MAIN SUCCESS SCENARIO</b>	<ol style="list-style-type: none"> <li>1. Start the iDART application</li> <li>2. Enter username</li> <li>3. Enter password</li> <li>4. Click 'Login' button</li> <li>5. Click the "General Admin" button</li> <li>6. Select the "manage iDART Users" button</li> <li>7. Enter the new user details: Username and Password</li> <li>8. Select the radio button applicable if the user will be working as a study worker or report worker</li> <li>9. Click Save</li> </ol>
<b>EXTENSIONS</b>	
<b>SUB-VARIATIONS</b>	8. This section is left blank as the user will be considered to be a pharmacist.
<b>Superordinate Use Cases</b>	Login
<b>Subordinate Use Cases</b>	
<b>Secondary Actors</b>	
<b>OPEN ISSUES</b>	<ol style="list-style-type: none"> <li>1. Only one instance of a user can be logged in at one point in time. That is, the system should not allow the same user details to be used at the same time.</li> </ol>

## UC601 - Issue an appointment date for a new patient

<b>Use Case Number</b>	UC601
<b>Use Case Name</b>	Issue an appointment date for a new patient
<b>Goal in Context</b>	Allow the logged in user to create a new patient and issue an appointment date to the patient
<b>Scope</b>	
<b>Level</b>	
<b>Precondition</b>	1. The User must be logged in to the system.
<b>Success End Condition</b>	The User has successfully created a new patient and an appointment date has been issued to the patient
<b>Failed End Condition</b>	The User has failed to create a new patient and to issue an appointment date to the patient
<b>Primary Actor</b>	User
<b>Trigger</b>	1. A new patient has visited the clinic
<b>Input</b>	The patient's details

<b>Main Success Scenario</b>	<ol style="list-style-type: none"> <li>1. Log into iDART</li> <li>2. Click on 'Patient Admin' Button</li> <li>3. Click on 'Add New Patient' Button</li> <li>4. Enter Patient's details</li> <li>5. Click on 'Create Initial Prescription' button</li> <li>6. Select Doctor and duration of prescription</li> <li>7. Click on 'Add Drug to Prescription' Button to add the drugs</li> <li>8. Once all the drugs are added, click on 'Dispense this Prescription'</li> <li>9. Click on the date by 'Next Appointment' field to select the date for the next appointment</li> <li>10. Click on 'Create Package' to dispense the drug to the patient</li> </ol>
<b>Subvariations (Alternate Course of Action)</b>	<p>2 - 3 If the user does not want to create a new patient, he can click on the 'Cancel' button to cancel the operation at each of these steps.</p> <p>4 - 8 If the user does not want to dispense the drugs to the patient, he can click on the 'Close' Button to cancel the operation at each of these steps.</p>
<b>Extensions</b>	N/A
<b>Superordinate Use Cases</b>	UC101
<b>Subordinate Use Cases</b>	N/A
<b>Secondary Actors</b>	N/A
<b>Open Issues</b>	N/A

## UC602 - Update an existing patient's appointment date

<b>Use Case Number</b>	UC602
<b>Use Case Name</b>	Update an existing patient's appointment date
<b>Goal in Context</b>	Allow the logged in user to update an exiting patient's appointment date
<b>Scope</b>	
<b>Level</b>	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. The User must be logged in to the system.</li> <li>2. The patient must exist in the system</li> </ol>
<b>Success End Condition</b>	The User has successfully edited an existing patient's appointment date
<b>Failed End Condition</b>	The User has failed to edit an existing patient's appointment date
<b>Primary Actor</b>	User
<b>Trigger</b>	<ol style="list-style-type: none"> <li>1. The appointment date for an existing patient needs to be updated</li> </ol>
<b>Input</b>	The patient's new appointment date

<b>Main Success Scenario</b>	<ol style="list-style-type: none"> <li>1. Log into iDART</li> <li>2. Click on 'Patient Admin' Button</li> <li>3. Click on 'Update an Existing Patient' Button</li> <li>4. Select Patient via Patient Search</li> <li>5. Click on 'Treatment Management' tab</li> <li>6. Click on 'Next App Date'</li> <li>7. Select new appointment date for the patient</li> <li>8. Click on 'Save' button</li> </ol>
<b>Subvariations (Alternate Course of Action)</b>	2 - 7 If the user does not want to update an existing patient's appointment date, he can click on the 'Cancel' button to cancel the operation at each of these steps.
<b>Extensions</b>	N/A
<b>Superordinate Use Cases</b>	UC101
<b>Subordinate Use Cases</b>	N/A
<b>Secondary Actors</b>	N/A
<b>Open Issues</b>	N/A

### **UC603 - Dispense directly to a patient who has been on an appointment**

<b>Use Case Number</b>	UC603
<b>Use Case Name</b>	Dispense directly to a patient who has been on an appointment
<b>Goal in Context</b>	Allow the user to dispense to an existing patient for a later pickup to ensure that no new appointments are created i.e. appointment date must not change
<b>Scope</b>	
<b>Level</b>	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. The User must be logged in to the system.</li> <li>2. The patient must exist in the system</li> <li>3. The patient must have been on an appointment</li> </ol>
<b>Success End Condition</b>	The User has successfully dispensed to a patient to and the correct appointment date as well as the correct visit date has been created
<b>Failed End Condition</b>	The User has failed to dispense to the patient and the correct appointment date as well as the correct visit date has not been created
<b>Primary Actor</b>	User
<b>Trigger</b>	<ol style="list-style-type: none"> <li>1. The patient has been on an appointment and needs medication to be dispensed to him directly</li> </ol>
<b>Input</b>	The patient's ID and new appointment date
<b>Main Success Scenario</b>	<ol style="list-style-type: none"> <li>1. Log into iDART</li> <li>2. Click on 'Stock &amp; Dispensing' Button</li> <li>3. Click on 'Make Up Drug Packages for Patients'</li> <li>4. Select Patient via Patient Search</li> <li>5. Click on 'Create Package' button</li> </ol>

<b>Subvariations (Alternate Course of Action)</b>	4- 7 If the user does not want to dispense to the patient, he can click on the 'Cancel' button to cancel the operation at each of these steps. If the user wants to change the prescription of the patient, he can click on the 'Update Patient Prescription' Button which can be found in the Prescription Information section on the screen.
<b>Extensions</b>	N/A
<b>Superordinate Use Cases</b>	UC101
<b>Subordinate Use Cases</b>	N/A
<b>Secondary Actors</b>	N/A
<b>Open Issues</b>	N/A

## UC604 - Dispense to an existing patient for a later pickup

<b>Use Case Number</b>	UC604
<b>Use Case Name</b>	Dispense to an existing patient for a later pickup
<b>Goal in Context</b>	Allow the user to dispense to an existing patient for a later pickup to ensure that the correct appointment date as well as the correct visit date has been created
<b>Scope</b>	
<b>Level</b>	
<b>Precondition</b>	1. The User must be logged in to the system. 2. The patient must exist in the system
<b>Success End Condition</b>	The User has successfully dispensed to the patient and the correct appointment date as well as the correct visit date has been created
<b>Failed End Condition</b>	The User has failed to dispense to the patient and the correct appointment date as well as the correct visit date has not been created
<b>Primary Actor</b>	User
<b>Trigger</b>	1. A package for the patient needs to be created for later pickup
<b>Input</b>	The patient's ID and new appointment date
<b>Main Success Scenario</b>	<ol style="list-style-type: none"> <li>1. Log into iDART</li> <li>2. Click on 'Stock &amp; Dispensing' Button</li> <li>3. Click on 'Make Up Drug Packages for Patients'</li> <li>4. Select Patient via Patient Search</li> <li>5. Select 'I am creating packages for later pickup'</li> <li>6. Select new appointment date for the patient</li> <li>7. Click on 'Create Package' button</li> </ol>
<b>Subvariations (Alternate Course of Action)</b>	4- 7 If the user does not want to dispense to the patient, he can click on the 'Cancel' button to cancel the operation at each of these steps. If the user wants to change the prescription of the patient, he can click on the 'Update Patient Prescription' Button which can be found in the Prescription Information section on the screen.
<b>Extensions</b>	N/A

<b>Superordinate Use Cases</b>	UC101
<b>Subordinate Use Cases</b>	N/A
<b>Secondary Actors</b>	N/A
<b>Open Issues</b>	N/A

## UC301 Add Participant To Study

<b>Use Case Number</b>	UC301
<b>Use Case Name</b>	Add Participant To Study
<b>Goal in Context</b>	Allow the user to dispense to an existing patient for a later pickup to ensure that no new appointments are created i.e. appointment date must not change
<b>Scope</b>	
<b>Level</b>	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. The iDART properties file must be edited so that the "CIDAstudy" option is true.</li> <li>2. The User must be logged in to the system and have the "Study Worker" role assigned.</li> <li>3. The SMS feature properties file must contain the correct settings to connect to mobilisr</li> </ol>
<b>Success End Condition</b>	User is able to connect mobilisr and add the participant to the desired campaign
<b>Failed End Condition</b>	Participant is not added
<b>Primary Actor</b>	Study Worker user
<b>Trigger</b>	1. Participant agrees to join study and signs consent form
<b>Input</b>	The patient's
<b>Main Success Scenario</b>	<ol style="list-style-type: none"> <li>1. Log into iDART</li> <li>2. Click on 'Add Participant to Study' Button</li> <li>3. Scan or search the selected patient to be added to the study</li> <li>4. Enter patients mobile number</li> <li>5. tick the "informed consent given" box</li> <li>6. select the campaign to add the user</li> <li>7. click add to study</li> </ol>
<b>Subvariations (Alternate Course of Action)</b>	4- 7 If the user does not want to dispense to the patient, he can click on the 'Cancel' button to cancel the operation at each of these steps. If the user wants to change the prescription of the patient, he can click on the 'Update Patient Prescription' Button which can be found in the Prescription Information section on the screen.
<b>Extensions</b>	N/A
<b>Superordinate Use Cases</b>	UC101 ;UC104
<b>Subordinate Use Cases</b>	N/A
<b>Secondary Actors</b>	N/A
<b>Open Issues</b>	N/A

## UC401 Remove Participant From Study

<b>Use Case Number</b>	UC401		
<b>Use Case Name</b>	Remove Participant from study		
<b>Goal in Context</b>	Allow the study worker to remove participants from a specific study		
<b>Scope</b>			
<b>Level</b>			
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. The iDART properties file must be edited so that the "CIDAstudy" option is true.</li> <li>2. The User must be logged in to the system and have the "Study Worker" role assigned.</li> <li>3. The SMS feature properties file must contain the correct settings to connect to mobilisr</li> </ol>		
<b>Success End Condition</b>		User is able to connect mobilisr and remove the participant from the desired campaign	
<b>Failed End Condition</b>	The user was unable to remove the participant from the campaign		
<b>Primary Actor</b>	User		
<b>Trigger</b>	1. Study has come to an end		
<b>Input</b>			
<b>Main Success Scenario</b>	<ol style="list-style-type: none"> <li>1. Log into iDART</li> <li>2. Click on 'Remove Participant From Study' Button</li> <li>3. Scan or search the selected patient to be removed from the study</li> <li>4. Click the "Remove from study" button</li> </ol>		
<b>Subvariations (Alternate Course of Action)</b>	3-4 If the user does not want to dispense to the patient, he can click on the 'Cancel' button to cancel the operation at each of these steps.		
<b>Extensions</b>	N/A		
<b>Superordinate Use Cases</b>	UC101 ; UC104;UC301		
<b>Subordinate Use Cases</b>	N/A		
<b>Secondary Actors</b>	N/A		

<b>Open Issues</b>	N/A
--------------------	-----

## CIDA Requirement Descriptions

### FR 601 - Issue an appointment date for a new patient

#### Description

Allow the user to create a new patient and issuing an appointment date for the patient

#### Business Rules & Error Messages

Business Rule	Business Rule Description	Error Message
BR601A	Patient must have been created	
BR601B	Patient must have had medication dispensed to him	
BR601C	Appointment date must be in the future. If the date is in the past the following error message should be displayed.	You can't set an appointment date in the past. Please select a date after today.

### FR602 - Update an existing patient's appointment date

#### Description

Allow the user to update an existing patient's appointment date in te update patient screen

#### Business Rules & Error Messages

Business Rule	Business Rule Description	Error Message		
BR602A	Patient must exist			
BR602B	Appointment date must be in the future. If the date is in the past the following error message sgould be displayed	Please select a date greater than today's date.		

### FR 603 - Dispense directly to a patient who has been on an appointment

#### Description

Allow the user to dispense directly to a patient to ensure that the correct appointment date as well as the correct visit date has been created

#### Business Rules & Error Messages

Business Rule	Business Rule Description	Error Message
BR603A	Patient must exist	
BR603B	Patient must have had medication dispensed to him	

BR603C	Appointment date must be in the future. If the date is in the past the following error message should be displayed		Please select a date greater than today's date.			
BR603D	Visit date must be equal to the date the patient visited the hospital					

## FR604 - Dispense to an existing patient for a later pickup

### Description

Allow the user to dispense to an existing patient for a later pickup to ensure that no new appointments are created i.e. appointment date must not change

### Business Rules & Error Messages

Business Rule	Business Rule Description	Error Message		
BR604A	Patient must exist			
BR604B	The user must dispense medication to the patient for a later pickup			
BR604C	Appointment date must be in the future			
BR604D	Visit date must be equal to the date the patient visited the hospital			

## FR301 Add Participant to Study

### Description

Allow the user to create a new patient and issuing an appointment date for the patient

### Business Rules & Error Messages

Business Rule	Business Rule Description	Error Message		
BR301A		A patient cannot be added to study if they already added		The patient you are trying to add to a study, has already been added to a different study.

BR301B		The mobile number entered in iDART must be in the format accepted by Mobilisr. No special characters and no spaces included in the number.		The mobile number of this participant is in the wrong format. Please remove all spaces and special characters. eg 0821234567
BR301C		A participant cannot be added to another campaign in the list if already added a CIDA campaign. A Participant can only partake in one CIDA campaign at a time.		The patient you are trying to add to a study, has already been added to a different study.

## **FR401 Remove Participant from Study**

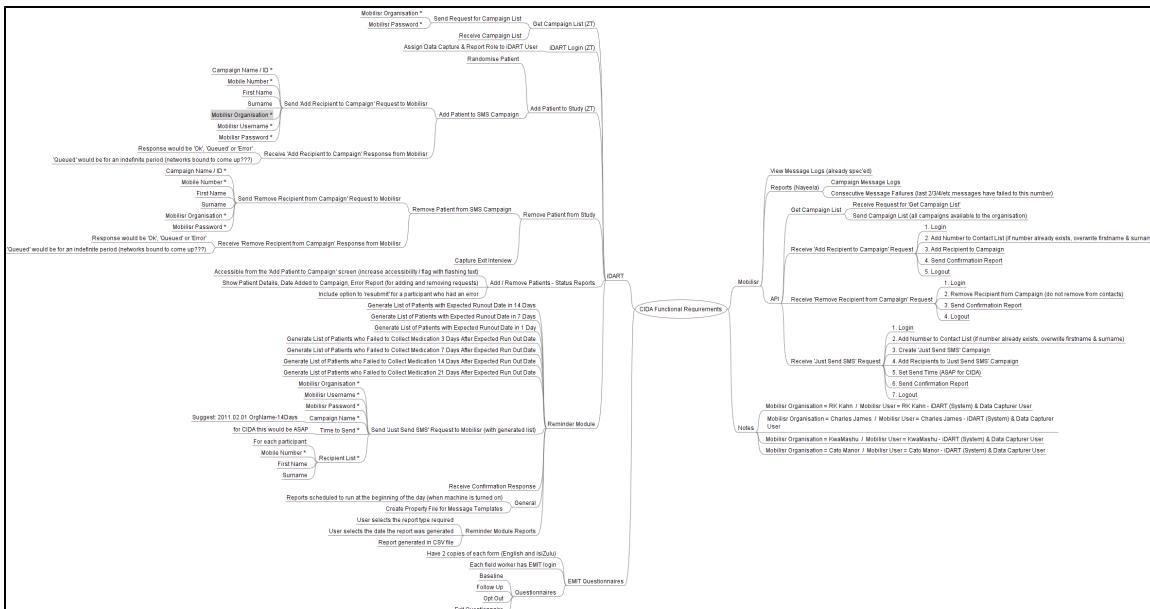
## Description

Allow the user to create a new patient and issuing an appointment date for the patient

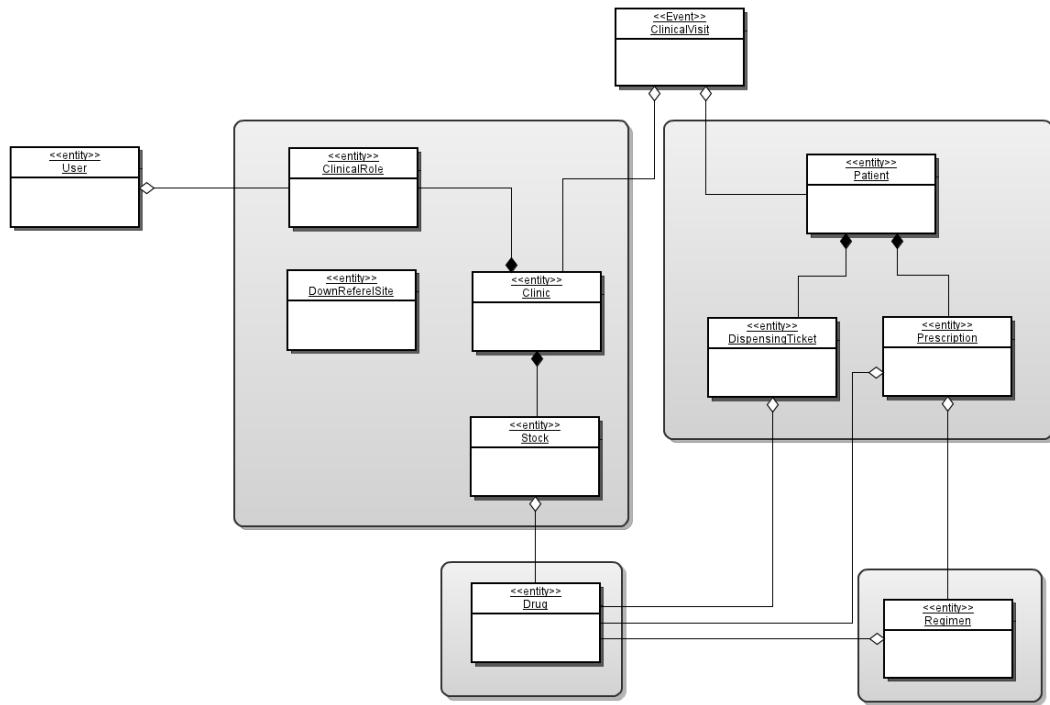
## **Business Rules & Error Messages**

Business Rule	Business Rule Description	Error Message				
BR401A	Only patients that are on an active CIDA study can be removed	The patient that you are trying to remove is currently not on any study	.			

## Mind Map - CIDA Functional Requirements



## Clinical Dispensing



## Glossary

### antiretroviral drug

Web definition:

- **Antiretroviral drugs** are medications for the treatment of infection by retroviruses, primarily HIV. When several such drugs, typically three or four, are taken in combination, the approach is known as **Highly Active Antiretroviral Therapy**, or HAART.

### clinic

Noun:

1. A place or hospital department where outpatients are given treatment or advice.

### clinician

Noun:

1. A doctor having direct contact with and responsibility for patients, rather than one involved with theoretical or laboratory studies.

### clinical assistant

Web definition: <http://www.hipro.com/MSL-205325-871/Ask-the-expert-What-is-the-definition-of-a-clinical-assistant.html>

- A clinical assistant is an individual qualified by academic education and clinical experience or other training to provide patient care services in a clinical or supportive role. CAs provide services only under the supervision of a member of the medical staff. CAs provide only those clinical services that are consistent with a written scope of care approved by the medical staff and senior management of the hospital. CAs are not members of the medical staff and are not privileged through the medical staff

### doctor

Noun:

1. A qualified practitioner of medicine; a physician.

## **down-referral**

Adhoc defintion taken from <http://isiafrica.net/?p=18>

- Down Referral is a mechanism to increase access to care for large numbers of patients requiring Anti-RetroViral (ARV) therapy. Down referral to primary health care sites often ensures that patients receive treatment within walking distance of their homes, reducing the financial burden associated with transport and loss of income for clinic visits. In South Africa PEPFAR-funded NGO [Right to Care](#) has worked with US biotech company [TherapyEdge](#) to develop a module for TherapyEdge's suite of HIV treatment products. The system maintains patient record continuity between the initiation site and the primary health care down referral site. The higher level prescribes the drugs. The lower level records the details of each patient visit: if the patient is stable the drugs are issued, if not the patient is referred up again.

## **down-referral site**

Adhoc definition taken from [http://www.righttocare.org/index.php?option=com\\_content&view=article&id=50&Itemid=85](http://www.righttocare.org/index.php?option=com_content&view=article&id=50&Itemid=85)

Mandatory features of a down-referral site:

- No doctor
- A nurse-run service: Nurse clinician or an ordinary nurse as the senior care provider dependent on services offered
- 'Task shifting': The allocation of tasks previously done by doctors or professional nurses to staff with less training (and lower salaries)
- No pharmacy involved : drugs are packed for patients at the initiation site and transported to the maintenance site
- Pharmacy storeroom to store drugs sent from initiation site

Optional features of a down-referral site:

- Facility for six monthly blood draws for monitoring
- Facility for interpretation of the blood drawn
- Either clinical review at every visit; or alternatively most visits are purely for drug collection, with clinical reviews at intermittent intervals.

## **drug**

Noun:

1. A substance that has a physiological effect when ingested or otherwise introduced into the body, in particular.

Synonyms:

- medicine, medicament, remedy, cure

## **episode**

Noun:

1. An event or a group of events occurring as part of a larger sequence; an incident or period considered in isolation.

## **hospital**

Noun:

1. An institution providing medical and surgical treatment and nursing care for sick or injured people.

## **infirm**

Adjective

1. Not physically or mentally strong, esp. through age or illness.

## **nurse**

Noun:

1. A person trained to care for the sick or infirm, esp. in a hospital.

## **outpatient**

Noun:

1. A patient who receives medical treatment without being admitted to a hospital: "attending a clinic as an outpatient".

## **patient**

Noun:

1. A person receiving or registered to receive medical treatment

## **pharmacist**

Noun:

1. A person who is professionally qualified to prepare and dispense medicinal drugs.

## **pharmacy**

Noun:

1. A store where medicinal drugs are dispensed and sold
2. The science or practise of the preparation and dispensing of medicinal drugs

## **prescription**

Noun:

1. An instruction written by a medical practitioner that authorizes a patient to be issued with a medicine or treatment.
2. The action of prescribing a medicine or treatment.

## **primary health care**

Web definition:

- health care that is provided by a health care professional in the first contact of a patient with the health care system.

## **regimen**

Noun:

1. A prescribed course of medical treatment, way of life, or diet for the promotion or restoration of health

## **stock**

Noun:

1. The goods or merchandise kept on the premises of a business or warehouse and available for sale or distribution

Verb:

1. Have or keep a supply of (a particular product or type of product) available for sale

## **Database Backup**

In order to provide backup facilities for iDART we have developed a small tool to perform the database backup and upload it to nkosi.cell-life.org.

There are two versions of the tool, one for Windows and one for Linux. The code for the is located here:

- Windows: <http://svn.cell-life.org/svn/internal/databaseBackup>
- Linux: <http://svn.cell-life.org/svn/internal/databaseBackupLinux>

There is also a pre-built version of the Windows tool on biko with an instruction manual: \\biko\cell-life.shared.idart\DatabaseBackup

## How it works

Both versions of the tool use the `pg_dump` command (installed on the machine) to dump the database. The tool then uses the modem to create a GPRS (or higher) connection and uploads the file (via `SCP`) to `nkosi.cell-life.org`.

On Windows the tool uses the `rasdial` command to establish the network connection (using an existing network connection setup in Windows). On Linux the tool issues AT commands to the modem to connect.

The settings for the tools are in a file called `backup.properties`.

## Functional Requirements List (v3.5X, Rewrite & CIDA)

### Current iDART functions

- Add a patient
- Edit a patient
- Add a clinic
- Edit a clinic
- Add a pharmacy
- Edit a pharmacy
- Add users
- edit users
- Add drugs
- edit drugs
- Add prescribers (doctors)
- Edit prescribers (doctors)
- Add drug groups
- edit drug groups
- Create prescriptions
- Update a prescription
- Merge duplicate patient records
- Capture patient test results (Stats Module)
- Create a package
- Create a package for later pickup]
- Track patient adherence via pill counts
- Add stock
- Remove stock
- Destroy stock
- Undo created package (should be Undo option for all actions)
- Scan created packages out to patients
- Return uncollected packages to pharmacy
- Perform stock take
- Down-refer patients
- Access down referal app
- Receive created packages from main clinic
- View stock on hand
- Reprint all labels
- Print labels
- Allow patient search
- Integrate with eKapa
- Integrate with OpenMRS
- Generates static reports
- Exports user defined data
- Round-up syrup values
- Detect broken drug packs
- Allow back button
- Reset screen via Clear button

- Navigate via shortcuts within screens

#### **Proposed iDART functions to be developed**

- **integrate communication service (SMS) - CIDA requirement**
- allow multiple prescriptions
- ability to adjust to various label sizes
- ability to add a drug with multiple pack sizes
- Edit reasons(user defined) (episode,prescription)
- Configure clinics for dispensing (will it be direct dispensing or for later pickup)
- Allow iDART to dispense acute medication (once off prescriptions)
- Edit a prescription without change prescription date(if not yet dispensed & password req)
- Edit dispensed record (replace an item for example)
- Generate user defined reports
- Customize labels - user defined in admin section
- Assign "Programme" to patient (instead of episode)
- Allow multiple "Programmes" for patients
- navigate to any section from any screen

#### **iDART Encore functions**

- **login functionality**
- Authentication and authorisation
- Auto logout on timer
- Audit log
- create/edit/delete stock groups
- create/edit/delete trade drugs
- create/edit/delete drugs
- create/edit/delete drug groups
- create/edit/delete active drugs
- create/edit/delete drug forms
- create/edit/delete drug types
- create/edit/delete dosage intervals
- edit simpledomain
- create/edit/delete users
- create/edit/delete roles
- create/edit/delete attributes
- create/edit/delete prescribers
- Stock management (receive, view and destroy stock)
- create/edit/delete patients
- create/edit/delete prescriptions
- allows multiple prescriptions
- advanced search functionality
- report view (although no reports have been created)
- Allows multiple drugs with the same name but different packsizes
- modular functionality

#### **CIDA - Generally there are three components to this project from a tech perspective:**

- 1. Mobilisr Server API: Currently in Mobilisr, external apps can't access Mobilisr so would require - allow external apps to access Mobilisr
- 2. Mobilisr Client API: Code on iDART side, independent from iDART that knows how to talk to Mobilisr.
- 3. GUI designs in iDART (this is dependent on which version of iDART is used) - only part that's specific to path

#### **Notes for CIDA Planning**

- CIDA: Research and BA meant to happen over the entire year, so may get to a point where the developers work on the rewrite while BAs/Research does needs analysis
- Keep in mind the ideas behind 'Programs' for the rewrite.

#### **Current iDART:**

- Add interfaces to iDART - new screens
- Additional features such as user based access
- Connectivity - system setup for new routers

- Risk is rewrite will never be completed

## **Getting started**

Step 1 - Install the necessary third party software

Step 2 - Retrieve a copy of the source code

Step 3 - Create a database

Step 4 - iDART development environment setup

Step 5 - Run iDART

Troubleshooting

## Step 1 - Install the necessary third party software

- Eclipse (<http://www.eclipse.org/downloads/>), including the subclipse or subversive plugin
- PostgreSQL version 8.2 or above (<http://www.postgresql.org/download/>).

## Step 2 - Retrieve a copy of the source code

Use Eclipse to checkout the source code into your workspace from this url:

<http://svn.cell-life.org/svn/idart/idart/trunk>

## Step 3 - Create a database

Having installed PostgreSQL you need to create a new database that iDART can use. The best way to do this is to run pgAdminIII (the graphical administration interface for PostgreSQL), connect to the database server and create a new database.

Alternativley you can use the *psql* command line client to connect to the database server and create a new database. Refer to the PostgreSQL documentation for more information.

### For versions less than 3.5.0

If you have checked out one of the iDART branches with a version less than 3.5 you will need to manually run the database scripts to create the database tables:

In the */path/to/project/metadata/database* folder you will find some scripts to create the tables and populate the minimal database. Run the sql script *idart\_3.0.0\_schema.sql* to create the tables. Then, run *idart\_3.0.0\_coredata.sql* to populate the database with the minimum required data. Then, to update to the latest schema version, run *idart\_3.0.0\_to-latest-sqldiff.sql*

## Step 4 - iDART development environment setup

Once you have checked out the source code and created a database you can follow these steps to run iDART:

1. Customise the build settings. Some of the build settings such as the database settings need to be customised for your local environment. To do this create a *local.properties* file in the project root (don't edit the *build.properties* file). Any properties in the *local.properties* file will override values in the *build.properties* file. Generally you will only need to customise the following properties:

### local.properties

```
# Database settings for running iDART
database.name=idart
database.username=postgres
database.password=postgres
database.host=localhost

# Database settings used when running unit tests from ant
database.name.test=idart-test
database.username.test=postgres
database.password.test=postgres
database.host.test=localhost
```

2. Having changed the database settings you can run the ant build (run the default task - *build*). This will compile the source code and the report files and will also create an *idart.properties* file in the project root directory that iDART will use when it runs (this properties file contains encrypted versions of the database settings as well as other general settings).

Note: If you change the database settings you will need to re-generate the *idart.properties* file with these new settings before running iDART. To do this run the *generateProperties* task in the ant build file.

## Step 5 - Run iDART

You are now ready to run iDART.

1. Locate the *org.celllife.idart.start.PharmacyApplication* class (located in the src folder). This class is the entry point for iDART. Right click on the class in the Package Explorer and select *Run As -> Java Application*.
2. You should see the iDART splash screen.
  - a. If you are running iDART 3.5 and above you will see the database setup wizard which will guide you through populating the database with the correct tables and data. This replaces manual running of the database scripts in versions lower than 3.5.
3. Log in with the default username and password: **admin : admin**

## Troubleshooting

See [Troubleshooting iDART](#)

### Postgresql

#### Setting up Postgres on Fedora

Setup and run postgres

```
$ sudo su - postgres
$ initd
$ postgres -D /var/lib/pgsql/data
```

Install an empty iDART database

```
$ createdb pharm
$ psql pharm
pharm=# CREATE ROLE pharm WITH SUPERUSER LOGIN PASSWORD 'pharm';
pharm-# \q
```

Test login of new user

```
$ psql --username=pharm --host=localhost --port=5432 --dbname=pharm
```

Edit /var/lib/pgsql/data/pg\_hba.conf in order to force the users to login with a password

local	all	postgres	trust
local	all	all	md5

### PSQL syntax

See all the tables in the database

```
pharm=# SELECT * FROM pg_catalog.pg_tables where tableowner='pharm';
```

## Troubleshooting

- Database issues
- Printing issues
- Scanner issues
- plpgsql Language not found in the Database
- java.lang.UnsatisfiedLinkError: no swt-gtk in swt.library.path, java.library.path

## Database issues

**Problem:** Cannot connect to the Database.

**Resolution:**

- After installing PostgreSQL, edit the postgresql.conf file found in C:\Program Files\PostgreSQL\8.3\data.<br/> In the Connections and Authentication section, replace <tt><nowiki>#listen\_addresses h3. 'localhost' </nowiki></tt>with <tt>listen\_addresses '\*'</tt>
- Make sure the database name entered during the iDART install is spelt exactly as it is in Pgadmin
- The database update script has been run in Pgadmin
- Make sure that you have entered the correct Database user name and password
- If the database is found on a network PC then enter the IP address of that PC in the Database server section

**Problem:** Patients that have been dispensed to the previous month cannot be found on the database

**Resolution:**

- Make sure that there is only one server running and enabled
- Make sure there is not another database with the same name

## Printing issues

**Problem:** The label printer is printing blank labels

**Resolution:**

- Make sure that the printer ribbon roll has been properly inserted

**Problem:** Unable to install the printer in Ubuntu

**Resolution:**

- A list of printers will be displayed once the printer has been detected, choose Generic and click Forward, select Raw printer and click forward.

## Scanner issues

**Problem:** The scanner cannot scan certain barcodes

**Resolution:**

- The PatientID cannot be shorter than 5 characters, otherwise the barcode that is printed will not be readable to the scanner.

## plpgsql Language not found in the Database

There are times when the **plpgsql** language needs to be installed in the database in order to run the scripts that we use.

Bellow is a post for the PostgreSQL site concerning the **createlang** program:<http://www.postgresql.org/docs/8.3/static/app-createlang.html>

This should be helpful for anyone installing the language, but bellow you should find a quick summary:

1. Run pgAdmin III and connect to the iDART database.
2. Open the query window.
3. Execute the following query:

```
createlang plpgsql;
```

4. You should have sucessfully installed the language by now... if not please refer above for a more concise description of this procedure.

## java.lang.UnsatisfiedLinkError: no swt-gtk in swt.library.path, java.library.path

If you receive this error it is because the wrong version of the swt library is in the project build path. To rectify this ensure that the build path has the correct swt jar for the operating system you are working on.

To change the jar file:

1. right click on the project and select properties
2. select Java Build Path in the properties dialog.
3. Click on the Libraries tab
4. Locate the swt library and click the remove button.
5. Click the Add JARs button and locate the swt-debug.jar file under idart-trunk/lib/swt-win or idart-trunk/lib/swt-gtk

## How to contribute

### Project SVN Repository

This project uses [Subversion](#) to manage its source code. Instructions on Subversion use can be found at <http://svnbook.red-bean.com/>.

#### Anonymous access

The source can be checked out anonymously from SVN with this command:\$ svn checkout <http://svn.cell-life.org/svn/idart/idart/trunk>

#### Developer access

Email a request to the [developers mailing list](#).

To commit changes to the repository, execute the following command to commit your changes (svn will prompt you for your password)\$ svn commit --username your-username -m "A message"

### Commit Comments

Code commit comments and ticket comments are linked in trac via key characters:

- Link to a specific revision with braces: [1234]
- Link to a specific ticket number with the pound sign: #123

#### Commits to trunk

- *when closing a ticket:* CLOSED - #123: description
- *if there is no ticket:* Fixing XXX

where "description" or "Fixing XXX" is a descriptive sentence about the error/function being corrected and 123 is the trac ticket id. The ticket id is optional, as there could be no trac ticket associated with the fix.

#### Commits to a branch

(include the branch name in the message)

- *when closing a ticket:* CLOSED - #123: in data\_export branch description
- Fixing XXX in data\_export branch
- Adding functionality X to data\_import branch

#### Merges

- Merging branch\_a to trunk [123]:[234]
- Merging branch\_a to trunk log:branches/branch\_a@123:234  
(this creates a Trac link to the changeset being merged)  
Where 123 and 234 are the revision start and end numbers for that merge.

#### Creating a branch

- Creating branch data\_import for rashid

## iDART 3.5 Functional Requirements

- *Clinic Management (FR100)*

**Design (BA)** refers to a BA designing the requirement. This can include process diagrams, GUI Mockups and Use Cases. When the BA is happy with the design, they mark the requirement as done and give their initials.

**Dev (Dev)** refers to a Developer developing the requirement. When the developer is happy with the requirement, they mark the requirement as done and give their initials.

**Test (BA/Dev)** refers to a developer / BA testing the requirement. When that person is happy with the testing, they mark the requirement as done and give their initials. If bugs are found, these are posted to Jira. Only once the requirement is completely finished can the requirement be marked as 'Tested'

### Clinic Management (FR100)

The **clinic** relates to both the down referral clinics, as well as the main clinic (at which patients collect their drugs directly from the pharmacy).

Req No	Req Name	Req Description	Process Diagram	GUI Mockup	Use Case	Additio nal Docs	Test Case		Design (BA)	Dev (Dev)	Test (BA)
FR101	Add New Clinic	Allow user to add a new clinic to the system. The user must be able to specify the name of the clinic (which will be displayed on reports and throughout the actual system, as well as the official details (from the national list of ARV sites in South Africa). These additional details are 'Facility Name', 'Facility	N.A.	GM101 Add Update Clinic		Excel spreadsheet of all ARV sites in SA			_____ (todo)	_____ (todo)	_____ (todo)

Type',  
'Sub-District',  
'District',  
'Province' and  
'Country'. Other  
details  
not  
contained in the  
national  
list  
include  
the  
facilities'  
telephone  
number  
and  
notes  
(which  
will be  
displaye

		d on the drug labels)						
FR102	Update View Clinic	Allow user to update an existing clinic in the system. The user must be able to select a 'Facility Name' from the national list of ARV sites in South Africa.	N.A.	GM101 Add Update Clinic				
FR103	Deactivate Clinic	Allow user to deactivate a clinic	N.A.	N.A.				

## iDART GUI Mockups

### GM101 Add Update Clinic

This screen is important for the BI future work for iDART (Siki working with Gillian to create BI application around iDART impact, specifically SA)

An Excel file of all ARV sites in SA is available.

All these sites need to be saved in a NEW database (lookup) table which would contain the following variables: Facility Name, Facility Type, Sub-District, District, Province, (possibly Country)

As more sites become accredited, the iDART Product Manager will provide an updated list which will be put into the standard iDART Upgrade Scripts.

Please remove the fields in the clinic table that are no longer needed - such as address1, address2, postalcode, (possibly?) province, etc.

Have to consider both existing and new sites & international sites

This is the current screen for 'Add New Clinic / Update Existing Clinic'

**iDART - Update Existing Clinic (logged in as admin)**

Select on icon      Update Existing Clinic

All fields marked with \* are compulsory

\* Clinic      Clinic Search

Location

Street

Province      ComboBox

City

Postal Code

Telephone

Notes

Save      Clear      Cancel

As before, the same screen would be used for both adding a new clinic and updating an existing clinic.

The user can either manually select their clinic by using the drop down options systematically (i.e. starting from 'Country', then 'Province', etc with the values being populated based on the item selected in the option above (i.e. all districts shown in Gauteng province)). Alternatively, they could search the 'National List of ARV Sites'.

The 'Search National List' would depend on a country being selected. If the user hasn't selected a country, inform them with an error message saying 'Please select a country first in order to load the national list of ARV sites' with an 'OK' option only. When they click 'OK' they are taken back to the 'Add / Update Existing Clinic' screen.

'Clear' clears all the text fields and combo boxes, disables all checkboxes besides the 'Country' one. It also sets all the text in the combo boxes to the default values of 'select a district' etc (as shown on this screen)

The 'Clinic Search' button loads the list of current clinics connected to this iDART pharmacy, not the complete list (i.e. its behaviour doesn't change).

National List of ARV Sites

Facility Name	Province
Hillbrow CHC	Gauteng
South Rand Hospital	Gauteng
etc	

Once the user has selected an ARV site from the list, then the user returns to the Add / Update Clinic with the details loaded. The 'Search National List' is still enabled however, so that if the user selected the wrong site, they could reload it by searching again.

## iDART CIDA Functionality

### CIDA Functionality

Since most of our sites will not be using the sms functionality, we have allowed this functionality to be turned off.

Thus by default the functionality is turned off and if you would like to use the functionality then you need only

set the cidaStudy property in the idart.properties file to true.

The CIDA functionality is configured in a properties file called the sms.properties file. This file allows the users to setup up to 3 appointment reminders and up to 3 missed appointment reminders.

Please note that if you only want to use 1 or 2 of the reminders, then leave the other values out of the sms.properties file.

These messages are automatically sent when a user logs into iDART.

iDART will try to schedule these messages every hour if they have failed to be sent the first time.

Once they have been successfully sent to Mobilis then iDART will stop trying to send them.

If for some reason the messages could not be sent to Mobilis on a particular day, during the next day these messages will be sent to the alerts table.

The following properties can be set:

**appointmentReminderValue1** - This is the number of days before an appointment that a patient would receive an sms

**appointmentReminderValue2** - This is the number of days before an appointment that a patient would receive an sms

**appointmentReminderValue3** - This is the number of days before an appointment that a patient would receive an sms

**appointmentReminderMessage1** - This is the sms message that the patients above will receive.

**appointmentReminderMessage2** - This is the sms message that the patients above will receive.

**appointmentReminderMessage3** - This is the sms message that the patients above will receive.

**appointmentMissedValue1** - This is the number of days after an appointment has been missed that a patient would receive an sms

**appointmentMissedValue2** - This is the number of days after an appointment has been missed that a patient would receive an sms

**appointmentMissedValue3** - This is the number of days after an appointment has been missed that a patient would receive an sms

**appointmentMissedMessage1** - This is the sms message that the patients above will receive.

**appointmentMissedMessage1** - This is the sms message that the patients above will receive.

**appointmentMissedMessage1** - This is the sms message that the patients above will receive.

**mobilisrurl** - The url of the deployed mobilisr which will be used to send the sms.

**mobilisrusername** - The username for an account on mobilisr which is authorised to send the sms.

**mobilisrpassword** - the password for an account on mobilisr which is authorised to send the sms.

**controlcampaignid** - This campaign needs to be created on Mobiliser to hold all the control patients. Once created, enter the id of the campaign in this field.

Some additional notes.

- If you put a Reminder/Missed value in the properties file and you forget to put in a message value then the patients will receive a blank sms. For example, you may set the appointmentReminderValue1 and forget to put in the appointmentReminderMessage1.

- If you put a message value in the properties file and you forget to put in the corresponding Reminder/Appointment value then it will just be ignored and those messages will not be sent as iDART will not know which patients to schedule them for.

- If you set 2 or more of the Reminder/Appointment values the same then the 2nd one will be ignored. For example, if you set both the appointmentReminderValue1 and the appointmentReminderValue2 to 2, the appointmentReminderValue2 will be ignored and those patients will only receive 1 message. Please note that you are allowed to set an appointment reminder value and a missed appointment value the same. For example, if we set both the appointmentReminderValue1 and the appointmentMissedValue1 as 3, then both messages will be scheduled.

## iDART CIDA Technical Doc

### **Reminder Module:**

The new functionality for this feature is contained in the following java classes:

- **Alerts.java** : This table is used to inform the study manager of any problems that occurred
- **MessageSchedule.java** : This table holds all the details to schedule the campaigns
- **smsSchedulerJob.java** : schedules campaigns once after user logs into iDART
- **smsRetrySchedulerJob.java** : Tries to schedule campaigns that failed to be scheduled on the day.

The reminder module is automatically started when a user logs into iDART. Currently there are 2 jobs that are scheduled. These jobs can be found in the

PharmacyApplication.java class in the runIDART() method. These jobs are called "smsReminder" and "smsJob".

The **smsReminder** job is a once off job that happens immediately and never repeats itself. This job is represented by the SmsSchedulerJob.java class.

This class looks at the setting in the sms.properties file and tries to schedule the sms's accordingly. Firstly, the execute method tries to schedule the

appointment reminders. To do this it first gets the list of patients in the study who have appointments matching the criteria. This is done through the

utility manager class called SmsManger.java. The execute method then calls the createAndScheduleCampaign method. This method does the bulk of the work

and starts by creating a Mobilisr JustSendSms Campaign. This is done using the mobilisr client api. The list of patients are then added as contacts in the

mobilisr camapign. Finally the campaign is sent to Mobilisr. This coincides with iDART writing these scheduled campaigns to the MessageSchedule Table.

This is done so that iDART has a record of which camapigns were scheduled successfully.

Lastly, the execute method checks if there are any messages in the MessageSchedule table that were not scheduled successfully on the day that it was scheduled to run. The execute method then sends these campaigns to the alerts table.

The **smsJob** job is schedule to run every 60 minutes. This job is represented by the SmsRetrySchedulerJob.java. This class performs similar functions to the SmsSchedulerJob.java class except that its execute method gets called every 60 minutes. Of course, if the SmsSchedulerJob.java class succeeded in scheduling the campaigns then this execute method will do nothing.

## iDART Encore Functional Requirements

[iDART Functional Requirements - Table of Contents](#)

- Auditing
- FR113 Perform Clinic Search
- FR300 Manage Stock
- FR400 Manage Patient Data
- FR500 Manage Programme
- FR700 Print Labels
- Generate Reports
- FR1010 Import Data
- **FR100 Configure iDART**
- FR600 Manage Prescription
- FR300 Manage stock
- FR400 Manage patients

**Design (BA)** refers to a BA designing the requirement. This can include process diagrams, GUI Mockups and Use Cases. When the BA is happy with the design, they mark the requirement as done and give their initials.

**Dev (Dev)** refers to a Developer developing the requirement. When the developer is happy with the requirement, they mark the requirement as done and give their initials.

**Test (BA/Dev)** refers to a developer / BA testing the requirement. When that person is happy with the testing, they mark the requirement as done and give their initials. If bugs are found, these are posted to Jira. Only once the requirement is completely finished can the requirement be marked as 'Tested'

### Auditing

Req No	Req Name	Req Description (Includes Business Rules)	Process Diagram	GUI Mockup	Use Case	Test Case	Design (BA)	Dev (Dev)	Test (BA)
FR401	View audit event	Allow user to view the details of an audited event. Selecting a row in the audit log should load the details for the selected item into the details panel.	N.A.	GM401 View audit event			— (todo)	— (todo)	— (todo)

### FR113 Perform Clinic Search

Req No	Req Name	Req Description (Includes Business Rules)	Process Diagram	GUI Mockup	Use Case	Test Case	Design (BA)	Dev (Dev)	Test (BA)
FR113	Clinic Search	System should allow user to search for a particular clinic					—	—	—

### FR300 Manage Stock

Req No	Req Name	Req Description <b>(Includes Business Rules)</b>	Process Diagram	GUI Mockup	Use Case	Test Case	Design (BA)	Dev (Dev)	Test (BA)
FR303	Receive stock from other facilities	Receive stock that was borrowed to a facility.					— —	— —	— —
FR320	Perform stock take	System should be able to open a stock take, capture counted stock, compare differences and adjust the stock levels based on the stock counted.					— —	— —	— —
FR331	View current stock level	System should be able to allow the user to view the current stock level on screen without having to print a report.					— —	— —	— —

FR332	Transfer Stock	System should allow the facility to add other facilities to the database and be able to transfer stock from the main clinic to other facilities added to the database.					<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
FR333	Receive uncollected packages	The system should be able to receive stock that was returned by another facility if the package was not collected by the patient. The stock should be added to the rest of the stock in the database					<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

FR341	Dispense Prescription directly to patient	The system should be able to dispense medication to a patient and subtract the medication from the stock on hand. When dispensing the system should be able to detect broken drug packs and round up syrups.					—	—	—
FR342	Create packages for later pickup	System should allow the user to create a package for a patient as per their prescription in advance so that the patient can collect the package at a later time or date.					—	—	—

FR343	Record package collection date	System should be able to save and record the date and time when the package has been collected by the patient whether it is at the main clinic or at a down referral clinic.							
-------	--------------------------------	--	--	--	--	--	--	--	--

#### FR400 Manage Patient Data

Req No	Req Name	Req Description (Includes Business Rules)	Process Diagram	GUI Mockup	Use Case	Test Case	Design (BA)	Dev (Dev)	Test (BA)
FR404	Merge duplicate patient records	System should be allow the user to take two patient records for the same patient and merge the information into one record using only one of the patient ID's as the primary patient record							

FR405	Edit Patient Events	The system should keep a record of what has happened to the patient by providing a list of reasons that the user can choose when a patient starts, changes or stops treatment at a facility.					—	—	—
FR408	View Patient history	The system should allow the user to view a list of previous packages and dates it was received on screen in table view.					—	—	—
FR411 &412	Add clinic data	The system should allow the user to save the patients clinic data such as their weight, CD4 count and whether they are pregnant etc					—	—	—

FR413	Adjust appointment date	The user should be able to change the appointment date provided by the system					
FR414	Capture Visit reason	The system should automatically detect whether a patient visit was scheduled or not based on appointment date and if unscheduled should have a reason recorded.					

FR500 Manage Programme

Req No	Req Name	Req Description (Includes Business Rules)	Process Diagram	GUI Mockup	Use Case	Test Case	Design (BA)	Dev (Dev)	Test (BA)
--------	----------	--	-----------------	------------	----------	-----------	-------------	-----------	-----------

FR501	Add patient to Programme	A patient has to belong to a programme in iDART. There are 4 types of programmes viz HIV, TB, HIV/TB, General. Patients that are on the General programme cannot receive ART or TB treatment.					<span style="color: red;">— (todo)</span>	<span style="color: red;">— (todo)</span>	<span style="color: red;">— (todo)</span>
FR502	Add clinic to programme	Each Programmes is conducted by a specific clinic and that should be indicated with a drop down list for the user to choose which clinic the programmes is held at.							

## FR700 Print Labels

Req No	Req Name	Req Description (Includes Business Rules)	Process Diagram	GUI Mockup	Use Case	Test Case	Design (BA)	Dev (Dev)	Test (BA)
--------	----------	--	-----------------	------------	----------	-----------	-------------	-----------	-----------

FR700	Print labels	System should be able to connect to a label printer and print drug labels, patient labels, package labels, summary labels for multiple label sizes.						—	—	—
-------	--------------	---	--	--	--	--	--	---	---	---

#### Generate Reports

Req No	Req Name	Req Description (Includes Business Rules)	Process Diagram	GUI Mockup	Use Case	Test Case	Design (BA)	Dev (Dev)	Test (BA)

#### FR1010 Import Data

Req No	Req Name	Req Description (Includes Business Rules)	Process Diagram	GUI Mockup	Use Case	Test Case	Design (BA)	Dev (Dev)	Test (BA)
FR1010	Import existing iDART data	The system should be backward compatible with previous versions of iDART and allow all the information captured in the previous versions to be imported into the database.					—	—	—

#### FR100 Configure iDART

---

Req No	Req Name	Req Description (Includes Business Rules)	Process Diagram	GUI Mockup	Use Case	Test Case	Design (BA)	Dev (Dev)	Test (BA)
FR101	Login User	System should allow user to Login to the system and perform actions as outlined by the permissions assigned.							
FR102	Logout user	System should allow the user to log off the system by clicking on the logout button at any time.							
FR103	Logout user automatically	The system should automatically log out the user if no activity has taken place after a certain amount of time (configurable).							
FR121	Add new user	System should allow the admin or any user that has the required permission to add new users to the system.							

FR122	Update users details	A user should be allowed to edit details that were entered when creating the account. This includes username and password.							
FR123	Deactivate user	The system should allow the administrator to deactivate users no longer using the system. They should no longer be visible however there will be a record of transactions still on the system.							
FR131	Add new role	Admin user should be allowed to add new roles with various permissions as needed by the organisation							

FR132	Update user role	The administrator should be allowed to update permissions for the various roles created							
FR133	Delete a user role	The system should allow the administrator to delete unnecessary roles that are not applicable to the organisation							
FR141	Add attribute (data field) to person type	Admin user should be allowed to add more data fields when adding new patients or prescribers .							
FR142	Edit attribute details	Admin user should be allowed to edit an attribute that was created. Eg change the data type from boolean to numeric etc							

FR143	Delete attribute	Admin should be allowed to remove attributes that have been created earlier						
FR151	Add a prescriber	Admin user should be allowed to add doctors or nurses that prescribe treatment to patients						
FR152	Edit prescribers details	Admin should be allowed to edit the details of the prescribers that were added previously						
FR153	Deactivate prescriber	The admin should be allowed to deactivate a prescriber from the users display although it should still be in the database for auditing purposes						

FR161	Add user-defined reasons	Allow iDART users with the correct permissions to add various reasons to the drop down lists in the prescription screen and episode changes							
FR162	Delete user-defined reasons	Reasons that were created by a user can be deleted by a user with the relevant delete rights							
FR171	Record user transactions	When a user logs into the system any major transaction should be recorded for auditing purposes.							
FR180	Create permissions table	The system should have list of permissions that can be assigned to a user role.							

FR190	Add new clinics	The user should be allowed to add more clinics that the system will be interacting with whether it is dispensing medication , down-refer ring patients or transferrin g stock							
FR191	Configure clinics for dispensing	System should allow the user to determine what will be their default method for dispensing treatment to patients ie will it be direct or for later pickup							
FR192	Edit clinic details	The user should be allowed to edit the main clinic details and any other clinics that were created eg address and telephone number.							

FR193	Deactivate clinic	The user should be allowed to deactivate any other clinics where interactions have stopped. This will remove the clinics from the view so as not to select them.						
FR194	Add a pharmacy( stock group)	Allow the admin user to add a stock group (Pharmacy ) to allow the user to dispense from different stock.						
FR195	Edit a pharmacy( Stock group)	The admin user should be allowed edit the name of the pharmacy and allow it to be selected as a default or to remove the pharmacy from the list of created pharmacies.						

FR	Add an active drug	The system should allow the admin user to add new active drugs to the database which will be used when adding new drugs							
FR	Edit active drug	Admin user is able to change detail that were entered when the active drug was added. This will also allow the admin user to remove the drug from the list.							
FR	Add drug	The user should able to add any new drugs that were created to the system. This would include the user being able to create multiple pack sizes for each drug. <b>Every ARV drug will consist of at least one active drug</b>							

FR	Edit drug details	User should be able to edit the details of the drug <b>(Can only be done if the drug has not been dispensed yet)</b>							
FR	Delete drug	User should be allowed to delete the drug from the database							
FR	Add Drug category	Users can create categories that inform the pharmacist what type of illness/disease it treats. Eg ARV , TB , etc							
FR	Delete Drug category	A category can only be deleted if no drug belongs to it.							
FR	Add drug form	Allow users to add a drug form that is not in the list							
FR	Edit created drug form	Allow the user to edit the drug form that was created by the user							
FR	Delete drug form	Delete the drug form that was created							

FR	Add drug regimens	Allow the user to create new regimens for drug groups						
FR	Edit drug regimens	User is able to change the name and description of the regimen. (If regimen is changed then all drug groups associated to that regimen should automatically updated even for passed dispensing)						
FR	Delete drug regimens	User should be allowed to remove drug regimens. (drug groups associated with the drug regimen being removed is then given the default "Mixed" regimen)						

FR	Create Drug groups	User is able to create a group of individual drugs which is assigned to a regimen. This allows for speedy creation of prescriptions.						
FR	Edit a drug group	User should be allowed to change drug groupings as indicated by the regimen. eg DoH decides that a 1st line regimen should change, then the pharmacist can edit the group accordingly.						
FR	Delete Drug group	If a drug group is no longer used by a facility then the user is able to delete the group from the database						

FR	Configure labels	User should be allowed to choose which label sizes are used by the facility and to add the clinic details to the label header							
----	------------------	---	--	--	--	--	--	--	--

#### FR600 Manage Prescription

Req No	Req Name	Req Description (Includes Business Rules)	Process Diagram	GUI Mockup	Use Case	Test Case	Design (BA)	Dev (Dev)	Test (BA)
FR601	Create a Prescription	System should allow user to create multiple prescriptions for patients							
FR602	Stop a Prescription	System should allow user to stop the patients prescription if the patient receives a new one before the prescription expires.							

FR603	Delete a prescription	A prescription should be able to be deleted if no packages has been dispensed to the patient yet. This is mostly in the event of user errors when capturing a prescription.							
FR604	View/ update a prescription	The user is able to view all the current prescriptions for that patient. A table is available to view the previous prescriptions for that patient. ( <b>P</b> ossibly create a button to view previous prescriptions that will display the previous prescription screen with the renew button next to each prescription)							

FR300 Manage stock

---

Req No	Req Name	Req Description (Includes Business Rules)	Process Diagram	GUI Mockup	Use Case	Test Case	Design (BA)	Dev (Dev)	Test (BA)
FR301	Capture Stock	The system should be able to add the amounts of physical Trade drugs received by the facility.							
FR310	Destroy stock	The system should allow the user to destroy stock. This should be accompanied with a reason as to why it is being destroyed eg expired stock							

## FR400 Manage patients

Req No	Req Name	Req Description (Includes Business Rules)	Process Diagram	GUI Mockup	Use Case	Test Case	Design (BA)	Dev (Dev)	Test (BA)
--------	----------	--	-----------------	------------	----------	-----------	-------------	-----------	-----------

FR401	Add new patients	The system should be able to add patients to the database which will ask for the folder number, name surname sex as compulsory fields. view mockup of screen for more input							
FR402	Update patients details	User should be able to make changes to patient details							
FR403	Deactivate Patients	If a patient is no longer receiving treatment at the facility then they may be deactivated from the system so that they are not visible during patient searches. They will still be displayed in reports during a given period.							

## iDART Feature Stories

As a Clinic Administrator

I want to track the stock that is dispensed to patients

So that I can have visibility on stock loss

In order to track the stock dispensed to patients

As a pharmacist

I need to

As a pharmacist

I want to dispense ART to a patient

So that they can increase their CD4 counts

In order for me to dispense ART to a patient

As a pharmacist

I need to create a drug package for that patient

In order for the pharmacist to create a drug package for a patient

As a doctor

I need to create a prescription for that patient

In order for the doctor to create a prescription

As an administrator

I need to register the doctor on the system

In order for a doctor to create a prescription for a patient

As an administrator

I need to register that patient on the system

In order for the pharmacist to create a drug package for a patient

As an administrator

I need to add stock to the stock center to draw from

In order for me to add stock to the stock center

As an administrator

I need to register the stock center on the system

In order for me to add stock to the stock center

As an administrator

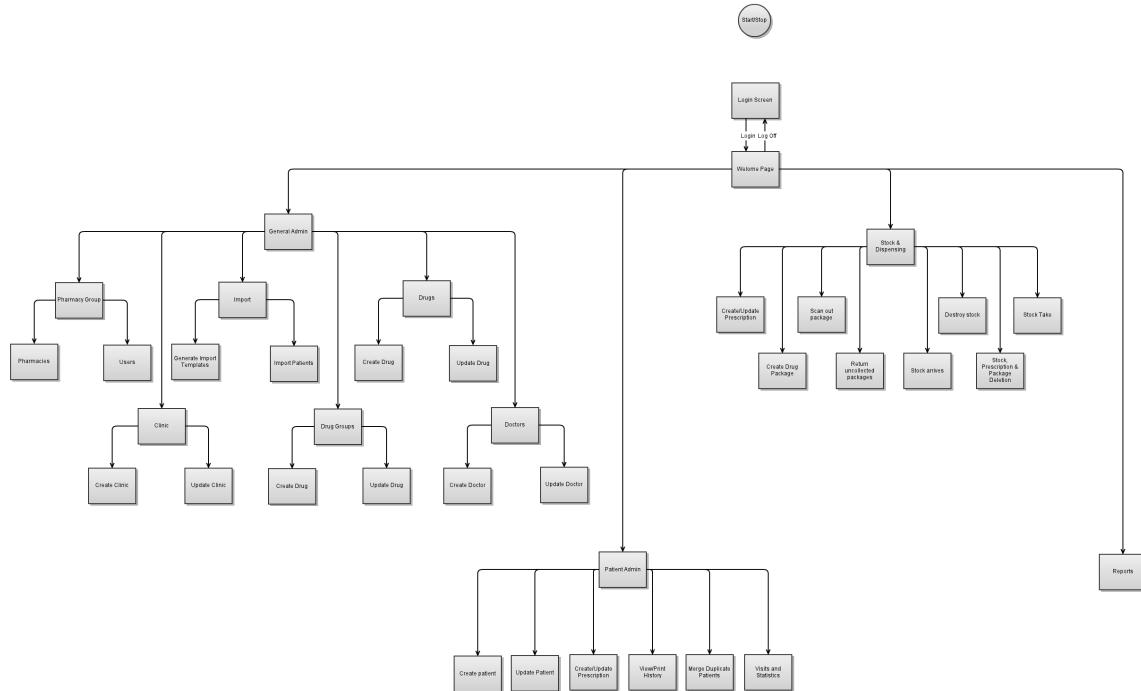
I need to register a drug on the system

In order for me to register a drug on the system

As an administrator

I need to register a drug compound on the system

## **IDART GUI Map**



## iDART Integration

### eKapa

- Custom connector in iDART that calls stored procedures in the eKapa database.
- Some procedures are called based on user interaction (e.g. patient search) and some are executed periodically (e.g. sending dispense information).
- Supports
  - Retrieve patient demographics (triggered by the user)
  - Send dispensing information (triggered by user dispensing to patient)
  - Send pill count information (triggered by user capturing pill count info during a dispense transaction)
  - Delete dispensing information (triggered by user deleting dispensing info)
  - Delete pill count information (triggered by user deleting pill count info)
- see [Ekapa.java](#), [EkapaSubmitJob.java](#), [StoredProcs.java](#)

### Data transferred between iDART and eKapa

#### Patient

- first name
- last name
- date of birth
- sex
- PAWC patient ID
- national ID
- home number
- work number
- cell number
- home address 1
- home address 2
- home address 3
- post code

#### Dispensing info

- reference id - for updating info later (iDART database ID)
- pawc number (patient ID)

- trade name
- trade strength
- chemical name
- trade pack size
- dispensed quantity (in dosing units)
- date dispensed
- repeat number
- units per dose
- dose units (tablets etc)
- dose route (per mouth)
- interval (3 times per day)
- duration in days
- special instructions (with meals)
- indication (for pain)
- date created
- user - name of user who created

## **OpenMRS**

- No changes to iDART code
- Use MIRTH as man in the middle
- Prescription info sent from OpenMRS to MIRTH in HL7 format
- MIRTH inserts the prescription into iDART database
- No data sent from iDART back to OpenMRS

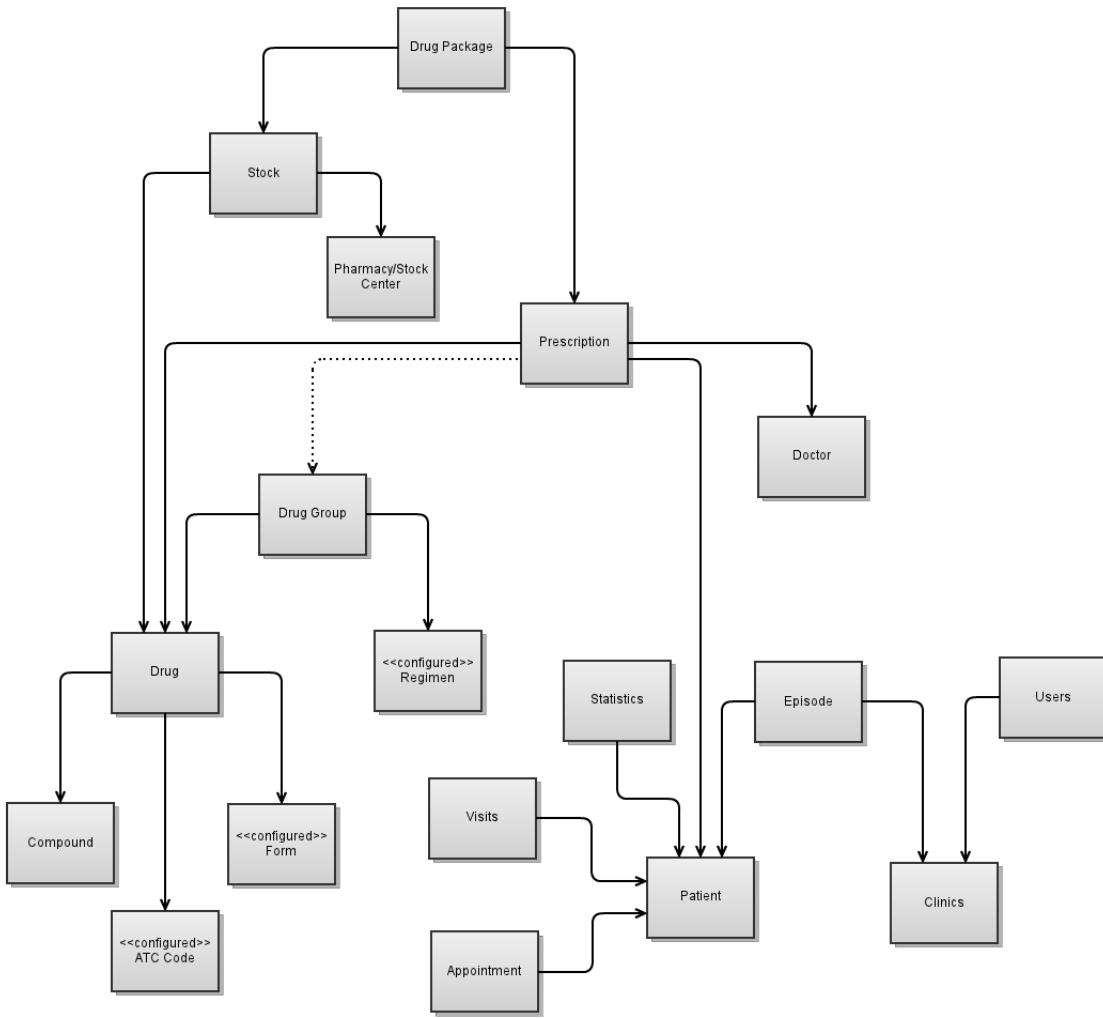
## **ieDEA / Tier.net**

This is a static export which can be imported into the Tier.net M&E system. It is run manually from the reports screen and produces a single XML file for each patient with data from their complete record.

See <http://www.iedea-sa.org/>, <http://tieredstrategy.wordpress.com/>

The data model for the export is in a separate project in the root of the iDART SVN: [org.iedea.sa-data-api](http://org.iedea.sa-data-api)

## **iDART Managed Entities**



As a pharmacist

I want to create

## iDART Reports

### Patient Reports

#### Patient History Report

Gives basic demographics of the patient and lists the prescriptions and the packages collected for each prescription.

#### Patient Defaulter Report / Missed Appointments Report

A report is generated for a particular day listing all patients that have missed their appointment dates. The user is able to specify how late a patient may be before being considered a defaulter.

#### Episodes Started and Ended Report

The User is able specify if s/he would like a report that shows all the patients that have started or have stopped treatment at the clinic.

The Episodes Started report provides the user with patient details such as patient ID , name , date started and the reason for starting at the clinic.

The Episodes Ended report provides the same details as the started report as well as the End date and End Reason only of those patients that have stopped treatment at that clinic.

The user is able to specify which date period he would like to report on.

## **Stock Reports**

### **Monthly Receipts and Issues Report**

The report is an indicator of how much drugs were received and how much were dispensed or destroyed within a one month period. It gives stock balances at the beginning and end of the month.

### **Daily Dispensing**

This report gives a general idea of how many patients visited the clinic and how much drugs was dispensed for that day. It does not provide and indication of which drugs were dispensed only the amount of ARVs and side treatments for that day.

### **Stock Take Report**

Allows you to check what the outcome of a stock take was. The information would tell the user how much was counted of a particular drug and what the difference was from what the system recorded. The user is allowed to select which previous stock takes s/he would like to report on.

### **Monthly Stock Receipts Report**

The report gives a detailed information on each drug received. It lists the batch number, the manufacturer and expiry date.

### **ARV Drug Usage Report**

This report provides information on how much of a particular drug was dispensed within a given time period. The user can only report on 11 drugs at a time.

## **Clinic Management Reports**

### **Packages Awaiting Pickup**

This report is only beneficial if the clinic does pre-packaged prescriptions as it will report on all packages that have been prepared for patients who will be picking up their packages at a later date. It is able to provide the package ID, the patient name and the date it arrived at the clinic and when the patient is expected to collect it.

This report allows you to see what patients have not picked up their packages yet and how late they are for the pickup.

### **Packages Received at Clinic**

This report can be used to confirm whether a package that was pre-packed for a patient arrived from the main pharmacy. It displays the date it was received, the package ID, the patient name and the patient ID.

### **Package Created Report**

This report is used as a confirmation of which packages were created during a specified period.

### **Package Collections Report**

This report displays all the patients that have picked up their pre-packaged drugs. It provides the date collected, patient details and package details. The user is able to specify the range that iDART needs to report on.

### **Packages Leaving Report**

This report lets you know what date a package was scanned out of the pharmacy to be delivered to a down referred clinic. It provides a list of all packages that were scanned out during a specific period.

### **Patients Expected on a Day**

The report provides names and patient IDs of all the patients that have appointments for a given day.

# M & E Reports

## Transaction Log Report

The Transaction Log Report records all transactions on iDART that would require the users password. Transactions such as creating a new user, deleting a package or prescription, destroying stock and merging patients. This report allows you to see who authorised those transactions.

## Prescribing Doctors Report

Generates a report that displays the number of prescriptions that were added or updated during a given period. It lists the date, name of doctor, reason for updating/creating a prescription as well as what was prescribed.

## Clinic Indicators Report

The clinic indicator report indicates how many patients are actively visiting the clinic. It provides statistics on how many patients are being treated, how many were initiated, how many patient visits and how many are no longer being treated by the clinic. The statistics generated are for a period that has been selected by the user.

## PEPFAR Report

The statistics generated by the PEPFAR report are related to how many patients are currently on a specific regimen. It also provides a summary of patients that are on ARV treatment and the reasons why patients are no longer on ARVs.

## iDART Roadmap

### iDART 3.5 (Q1 2010)

- Move clinic from prescription to packages
- Display active user on screen
- Patient name on package label
- Refactor label printing API
- Batch number on summary label
- Re-print most recent labels
- Drug batches dispensed report
- Stock in hand screen
- Changes to Episode reports
- Generate clinic indicator for ALL clinics

## References

### Change log

## Change Log

### Version 3.4.3

02 December 2009

- Fixed Stock Take Crash for initial stock take
- Allow user to change the clinic name
- Allow multiple undispensed packages
- Set main clinic as default clinic on login page
- Prevent users from creating duplicate clinics
- Allow access to clinic to all users when a new clinic is created
- Added import template and import Readme file to installer
- Removed chemical compound didanosine from drug tenofovir
- Added Regimens 1C and 1D to simple domain
- Added Activation reason "down Referred" to simple domain
- Improved imports to consider Clinics, closed episodes and province.
- Increased bar code width to enable scanning
- Fixed the incorrect adherence in the patient history report

- Added prescription export
- Fixed the error messages for multiple uncollected packages
- Users not allowed to create a clinic with duplicate name
- Removed the scan into clinic step in the down referral model
- Disable the inactive radio button when adding a doctor

## **Version 3.4.2 (internal release)**

### **Version 3.4.1**

12 June 2009

- removed clinic restriction on users from all places except login
- FIXED #1395 (idart.properties file
  - clean up (for ...)
- added linux local backup script backup.sh
- include schema scripts in install
- upgraded iText library from 1.3.1 to 2.1.0
- added "at clinicname" to episode startnotes when down referring patient
- mark appointments as inactive when patient comes early
- added manage down referral admin screen
- added down referral dialog
- added drug combination csv report
- fixed discrepancies between ClinicIndicator report and PepfarReport

## **Version 3.4.0**

17 April 2009

- inclusion of CD4 module code into iDART core
- removed Weight and WHO Stage from Prescription interface since they are now catered for by the CD4 module
- improved the demographics import

## **Version 3.3.2**

26 January 2009

- display accumulated drugs on drug labels
- redo-single-item: summary label still shows deleted items
- printing emergency labels causes crash if fields left blank
- auto round-up of drug quantities (configurable via properties)
- find latest postgres version on windows (or use value supplied in properties)
- property file to disable printing of drug labels (in case of no printer)
- added patient attributes to data export

## **Version 3.3.0**

21 August 2008

- Beta Release

### ***Stock now belongs to a pharmacy (previously belonged to a clinic)***

This is one of the biggest changes in iDART 3.3. In previous versions, you were only able to use the down-referral / pre-packaging module if you had separate stock for each of your clinics. That is, when you received stock, you would specify which clinic this was for, and only patients who went to that clinic to collect their drugs, would be allowed to get drugs from that particular set of stock.

Now in iDART 3.3, we have introduced the notion of having multiple 'pharmacies'. The clinics now have no relation to physical stock and any patient can receive medication from any stock that has been received in the system.

The 'Facility' relates to the physical pharmacy (it's address, responsible pharmacist, etc.) and it is this information that is shown on all the drug labels. You are only able to have one 'facility' per iDART installation, and this can be configured in 'General Admin -> Manage Pharmacies' (bottom half of the screen)

The 'Pharmacy' in iDART is related to where stock is physically held. Normally this is the same as the actual facility, but in some rare instances, you may have separate stock for certain groups of patients. This is normally related to clinical trials, or when drugs are sponsored by certain NGOs for certain patients. You are able to have more than one 'pharmacy' and this is configurable in 'General Admin -> Manage Pharmacies'(top half of the screen). From this screen you can also say which of these 'pharmacies' is the preferred one.

When you are dispensing ('Make Up Drug Packages'), you are able to select which pharmacy you would like to take your stock from. Your 'preferred' pharmacy is loaded by default, but this can be changed on this screen.

And, when you receive stock, you need to specify to which pharmacy this stock will belong.

### **Paper-Based Down-Referral**

This is a new module that was developed for ARV pharmacies that down-refer to smaller clinics and also then pre-package for these patients. It is intended to be used at facilities where there is no reliable internet connection. The process of tracking the packaged drugs is then done using a paper trail.

- Question added to installer to configure for this. User selects 'Use electronic communication between pharmacy and clinics' or 'Use paper-based communication between pharmacy and clinics' and this sets offerDownReferralFunctionality in idart.properties accordingly.
- New GUI showing table of created packages that have left the pharmacy (accessible from 'Scan out Packages to Down Referral Clinic')
- New report called 'Patient Collection Sheet' that can be printed out for each package. The nurse and patient would then sign this sheet and it would be used to back-capture the information.
- New report that replaces 'Packages Leaving Pharmacy' report.  
Note that depending on the value of offerDownReferralFunctionality (in idart.properties), either of these two reports will be available.
- The existing 'Packages Leaving Pharmacy' report shows 'Date Left', 'Package ID', 'Patient Name' and 'Patient ID'
- The new 'Packages Leaving Pharmacy' report shows 'Date Left', 'Patient ID', 'Patient Info', 'Drugs', 'Prescription Info', 'Package ID' (with barcode).

### **Other Features**

#### Multiple Pharmacies

- Pharmacy Information screen: minor formatting changes to accommodate multiple pharmacies functionality
- Removed question from installer as it is no longer necessary: "Do you currently have down referral clinics?" Previously this set the idart.property offerclinicfunctionality to true. It now automatically sets this variable to true.
- Packaging screen updated to allow selection of Pharmacy from which to use stock

#### Drugs

- Added ability to set "amount per time" property in a drug's standard dosage, to 0.5

#### Labels

- Changes to displaying "0.5" tablets on drug labels (now uses 'half' for tablets)
- Changes to Drug Labels: Next appointment date, quantity dispensed & accumulated, and compressed patient name shown
- Changes to Prescription Summary Label
  - next appointment date, quantity dispense and returned for each drug.
- Changes to Patient Info label: Added clinic. Combined first name and surname, gender written in full, date format for DOB.
- Custom drug label updated based on new drug labels format
- Ekapa and iDART drug labels combined

#### Stock Arrivals

- GUI Changes to Delivery Details (pre-populates manufacturer and cost based on previously captured data. Also updated to accommodate stock belonging to a pharmacy)
- Minor change to Stock Arrives GUI
  - screen closes after saving entries to database
- Costs added to Monthly Stock Receipts and Daily Stock Receipts reports

#### Data Export

- Additional data available for export: episode details and defaulters details (last collected drugs, last collected date, expected date patient will run out of drugs)
- Changes to logic in export name
- Export GUI updated to accommodate new info available for export

## Patient Admin

- Month combos updated to accept single letter months
- Package ID length now limited
- Change to logic of Patient Merge: Second patient is now deleted
  - deleted data recorded in log
- Inform users if they are closing an episode for a patient, but this patient still has an uncollected package

## General

- Minor formatting change to welcome screen (added shortcuts for the 6 screens
  - Alt + shortcut
- Version numbers added to Login screen and error screens
- Added catch-all error window for unhandled runtime exceptions
- Added ability to download and install minor updates from online update server

## Prescriptions

- Minor change to Prescription GUI
  - doctor list sorted alphabetically

## Reports

- New report called 'Episodes Started or Ended'. The user can specify date range as well as two columns for ordering.

## Version 3.1.8

28 October 2008

- Official Release

- Added stock returned/destroyed to report #835.
- Added ability to read package return reasons from database
- Updated installer to include schema and coredata scripts
- Bug fixes
  - #1131. enabled datePicker so it can be used for the report
  - #1113. Changed the wording in error message
  - #1127. Loaded patient details when coming from prescription screen
  - #1130. Fixed the initial loading information for packdate
  - #1137. Enabled the arvStartDate button for new patients
  - #1137. Cleared blank rows from table when user closes drug selection screen with 'x' button
  - #1131. Fixed date picker to allow date change
  - #1108. fixed update patient problem
  - #1128 (Scan out packages to remote clinic
    - iDART crashes)
  - #1125. Fixed dispensed quantities in package tracking report
  - #1113 (Dates of scanning of packages
    - Check Date is after packdate / dateleft / datereceived etc)
  - #1118. Added listener to clear table row when the user closes the add stock screen
  - #1116. Added keys to siteUser table.
  - #1117. Added title to message boxes
  - #1108. patient prescription not created when latest episode
  - #1115. changing heading wording on add to drug group.
  - #1106. Fixed order of prescriptions
  - #1100. Fixed duplicate drugs on patient history report
  - #1110. Fixed crash in package Return screen when returning drugs to stock
  - #1107. Fixed crash in Deletion screen when patient id with a space is entered
  - #1090. Saved correct appointment date
  - #1109. Updated the list of site for users
  - #1091. (RHRU Trip fixes and modifications)
  - Fixed iDART properties file. Was not reading all values
  - debug text removed from PackageReturn.java
  - other minor bug fixes

## Version 3.1.7

20 June 2008

- Official Release

- Added stock returned/destroyed to report #835.
- Added ability to read package return reasons from database
- Updated installer to include schema and coredata scripts
- Bug fixes
  - #1131. enabled datePicker so it can be used for the report
  - #1113. Changed the wording in error message
  - #1127. Loaded patient details when coming from prescription screen
  - #1130. Fixed the initial loading information for packdate
  - #1137. Enabled the arvStartDate button for new patients
  - #1137. Cleared blank rows from table when user closes drug selection screen with 'x' button
  - #1131. Fixed date picker to allow date change
  - #1108. fixed update patient problem
  - #1128 (Scan out packages to remote clinic
    - iDART crashes)
  - #1125. Fixed dispensed quantities in package tracking report
  - #1113 (Dates of scanning of packages
    - Check Date is after packdate / dateleft / datereceived etc)
  - #1118. Added listener to clear table row when the user closes the add stock screen
  - #1116. Added keys to siteUser table.
  - #1117. Added title to message boxes
  - #1108. patient prescription not created when latest episode
  - #1115. changing heading wording on add to drug group.
  - #1106. Fixed order of prescriptions
  - #1100. Fixed duplicate drugs on patient history report
  - #1110. Fixed crash in package Return screen when returning drugs to stock
  - #1107. Fixed crash in Deletion screen when patient id with a space is entered
  - #1090. Saved correct appointment date
  - #1109. Updated the list of site for users
  - #1091. (RHRU Trip fixes and modifications)
  - Fixed iDART properties file. Was not reading all values
  - debug text removed from PackageReturn.java
  - other minor bug fixes

## Version 3.1.6

27 May 2008

- Pre Official Release

- database clean up script
  - for removing test patients
- #577. Addded reasons to simple domain
- Fixed Bugs
- CLOSED: #1046. Fixed the updating of values in drugs table
- CLOSED: #1018. Fixed date problem in scan to patients screen
- CLOSED: #1048: Batch Info Screen: automatically calculating labels to ...
- CLOSED: #853
- CLOSED: #1023
- CLOSED: #1052. Added error message when user enters end date before start date
- CLOSED: #1050. cleared text field when successfully package scanned
- CLOSED: #1053. Fixed report date problem
- CLOSED: #1051. Fixed sorting problem for days late column
- CLOSED: #994
  - ARV Start date being properly removed at Package deletions.
- CLOSED: #1017
- CLOSED: #765. Fixed stock inconsistencies
- CLOSED: #1057, pick up date selection now properly working on scan packages ...

- CLOSED: #1032
  - Erroneous error message upon wrong specification for label ...
- CLOSED: #1064. needs to be tested thoroughly, not sure what this fix affects
- CLOSED: #1054
- CLOSED: #1059
- CLOSED: #1066
- CLOSED: #1069. Could not save packages
- CLOSED: #1068
- CLOSED: #1047 iDART crashes when deleting an undispensed prescription
- CLOSED: #835 fixed (added reason to report)
- CLOSED: #1068
  - pill count date properly implemented
- CLOSED: #1063 (Packages Collected by Patient Report
  - 'Current Issue' ...
- CLOSED: #1072 (Able to dispense more than in stock)
- CLOSED: #1073 (Can't destroy only loose pills)
- CLOSED: #1074 (Days since last pickup reset by changing weeks supply)
- CLOSED: #1057 (Scan Out Packages to Patient
  - Pickup date picker not ...
- CLOSED: #1067 (Patient History Report
  - Days Later value incorrect)
- CLOSED: #1068 (Pill Counts
  - Date of Count not being saved correctly ...
- CLOSED: #1075 Saved return reason to database
- CLOSED: #1076. Fixed prescriptionId

## Version 3.1.3

22 April 2008

- Maintenance release

- changes to generateScripts.py
- Fixed index problem in patient collection report
- force shutdown of all threads when iDART closes down
- fixed bugs
- #996. Cleared table correctly
- #992 Scan Out Packages to Patients
  - not setting the appointment date correctly
- #994 Package Deletions
  - If the package deletion violates the ARV Start Date rules, remove ARV Start Date
- #980 Adjusted label size to accommodate notes
- #930 Not able to change appointment data from patient editor
- #930 iDART crashing upon next appointment date validation
- #987 Fixed report date to reflect selected date
- #986 Improved usability of the packageIdScan textField
- Reworking patient merge to check for null dates when loading drug list in Gui, ticket #942
- #981 Packaging: User told iDART couldn't save package, but drug labels were printed
- #931 Added check for valid appointment date
- #953 Packaging GUI
  - Date of Last Pickup not getting updated when I change the 'Date Packed'
- #958, #933
- bug where Stock is not updated after StockLevel is deleted.
- #985 Added check to prevent users from dispensing to a patient with an uncollected package
- Fixed dispensing bug. Patients with no appointment could not be dispensed to

## Version 3.1.2

17 April 2008

- Maintenance release

- fixed stocklevel bug by adding session.flush

## Version 3.1.1

02 April 2008

- Maintenance release

- fixed package id bug where a '1' was being appended to the id
- CLOSED
  - #882: Add New Patient
  - pregnancy radio button disabled
- CLOSED
  - #883: Return uncollected packages
  - date of return date not ...
- CLOSED
  - #881: Add New Patient
  - Date of birth check no longer works
- CLOSED
  - #880: Add new patient
  - setting ARV Start Date causes iDART to ...
- CLOSED
  - #596: Bug: Editing previous episodes are saved, but text on ...
- CLOSED
  - #878: iDART ekapa version does not shut down correctly
- CLOSED
  - #877: Incorrect message and behaviour when dispensing to a ...
- CLOSED
  - #811: ARV Start Date
  - do validity checks when patient updates ...
- improved session management and class structure of Patient tabs
- CLOSED
  - #827: Pregnant Female patient
  - wording in error message to user
- CLOSED
  - #815: Log files for interrupting a report being printed
- REOPENED
  - #821: Error Log files: ERROR
  - The database version is out of ...
- CLOSED
  - #873: Prescription GUI
  - editable cells
  - need checks for this
- CLOSED
  - #813 Scan out packages to remote clinic
  - color of drop down list ...
- CLOSED
  - #871: Scan out packages to remote clinic
  - Date picker for date ...
- CLOSED
  - #870: Prescription shortcut from packaging screen
  - deleting ...
- CLOSED
  - #872: Return uncollected packages
  - stop date not formatted ...
- CLOSED
  - #860: Return Uncollected Packages
  - check for 'return drugs / ...'
- CLOSED
  - #830: Return Uncollected Packages
  - clicking on a package in the ...
- removed calls to system.out.println

## Version 3.1

12 March 2008

- Added cell editors to the prescribed drugs table on the prescription screen.
- This is particularly important for paediatric dosage amounts since they change frequently. It will also make the capturing process quicker, since users would be able to edit existing drugs without having to recapture them completely.
- Ability to return uncollected packages without deleting them.
- Packages can be marked as returned and their stock can either be discarded or added back into stock.
- changed version numbering
- LONG: major.minor.release DB database\_release Build svn\_revision (e.g. 3.1.0 DB 6 Build 2568)
- SHORT: major.minor.release.svn\_revision DB database\_release (e.g. 3.1.0.2568 DB 6)
- Added data export feature.
- Users are now able to generate csv files containing patient data.
- Remote Clinic functionality has been added.
- Users may now log on to iDART in 1 of 2 modes. These modes are the normal iDART mode and the new Clinic mode.
- Clinics can now receive drugs from a distribution point and dispense them to patients.
- Package tracking reports updated.
- Added an ART start date to all patients. Users may capture this field if it is known or can choose to ignore this field.
- Add Patient screen has been modified by using tabs. Users will be able to choose which information they would like to see or edit using the tabs.

## Version 3.0.4

12 March 2008

- Fixed bugs #567, #617, #618, #622, #624, #828

## Version 3.0.3

13 November 2007

- Refactor GUI classes
- Added pill count table to the scan out screen
- Episodes added to patient history report
- Re-work of stock take function
- New version of IzPack
- Adding drug to prescription brings up drug search by default
- Version number added to GUI
- Numerous bug fixes

## Version 3.0

- Ability to configure the chemical composition of drugs.
- Create/Update/Delete chemical compounds.
- Replace patient activation with episodes.
- A history of patient episodes.
- Patient history report available from the update patient screen.
- Record of previous patient identifiers used by a patient.
- Patient merging
- Recoding and reporting on adherence based on the pill counts.
- Access to patient's prescription from the dispensing screen.
- Screen to record pill counts without dispensing.
- Stock take
- Re-factored session management to improve stability

## iDART Test Cases

### iDART Test Cases

#### Test Cases for Reports (TC9XX)

- List of Test Cases still to be created

- Patient Reports (TC91X)
  - [TC911 - Patient History Report](#)
  - [TC912 - Episodes Started Report](#)
  - [TC913 - Episodes Ended Report](#)
  - [TC914 - Package Tracking](#)
- Stock Reports (TC92X)
  - [TC921 - Monthly Receipts and Issues](#)
  - TC922 - Daily Dispensing Totals
  - TC923 - Stock Take
  - TC924 - Drugs Dispensed
  - [TC925 - Monthly Stock Receipts](#)
  - [TC926 - ARV Drug Usage](#)
- Clinic Management Reports (TC93X)
 

Test Data for Package Movement Reports (downReferralModel = offline)

Test Data for Package Movement Reports (downReferralModel = online) - not tested yet

  - TC931 - Packages Created
  - TC932 - Packages Leaving Pharmacy
  - TC933 - Packages Received at Clinic
  - TC934 - Packages Collected
  - TC935 - Packages Awaiting Pickup
  - TC936 - Patients Expected On A Day
  - TC937 - Missed Appointments
- M&E Reports (TC94X)
  - [TC941 - Drug Combinations \(CSV\)](#)
  - TC942 - Episode Statistics
  - TC943 - Transaction Log
  - [TC944 - Prescribing Doctors](#)
  - [TC945 - PEPFAR Report](#)
  - [TC946 - Clinic Indicators](#)

## Monthly Stock Receipts report

### Purpose

Check that the user is able to generate a Monthly Stock Receipts report for each pharmacy correctly\*

### Prerequisites

1. Load the test database
2. Add the following drug groups to the test database
  - a. 3TC150 - D4T30 - NVP200 as Regimen 1
  - b. AZT300 - DDI400 - LPV/r 133/3 as Regimen 2
3. Add another pharmacy called Paed Drugs

### Test Data

- Add 10 units of 3TC150 - D4T30 - EFV600

#### **3TC 15mg (Same details need to be used for D4T and NVP)**

- Manufacturer: GSK
- Batch Number: abc123
- Expiry Date: January year(2 years after current date)
- Units : 10
- Pharmacy: main
- Arrival date: Current date

### Steps

1. Log in as admin

2. Click on the **Stock & Dispensing** button
3. Click on the **Stock Arrives at the Pharmacy** button
4. Add 10 units of 3TC 150mg
5. Add 10 units of D4T 30mg
6. Add 10 units of NVP 200mg
7. Click on the Save button
8. Click yes to view Stock Report
9. Navigate to the Reports section
10. Select **Monthly Stock Receipts Report**
11. Select the applicable month that the report needs to generate.
12. Check the report against the example attached to this test case.

#### **Notes & Questions**

## **TC93X - Test Data for Package Movement Reports (downReferralModel = offline)**

#### **Test Data**

Set the idart property **downReferralModel** to **offline**

#### **Patient 1**

##### *Patient 1 Details*

- Folder Number: DD001
- Patient Details: Donald Duck, 01 Jan 1980, Male
- Episode: New Patient on 01 Jan 2010 at **Main Clinic**

##### *Prescription 1 Details*

- Capture Date 01 Jan 2010
- Doctor: John Smith
- Duration: 1 month
- Drugs: 3TC 150 - D4T 30 - EFV 600 (1A-30)

##### *Dispensing 1 Details*

- Date Packed: 01 Jan 2010
- Package Contains: 1 month supply
- Dispensing Mode: **I am creating packages for later pickup**

#### **Patient 2**

##### *Patient 2 Details*

- Folder Number: DD002
- Patient Details: Daffy Duck, 01 Jan 1980, Female
- Episode: Transferred In on 01 Jan 2010 at **Clinic A**

##### *Prescription 2 Details*

- Capture Date 01 Jan 2010
- Doctor: John Smith
- Duration: 1 month
- Drugs: 3TC 150 - D4T 30 - NVP 200 (1B-30)

##### *Dispensing 2 Details*

- Date Packed: 01 Jan 2010
- Package Contains: 1 month supply
- Dispensing Mode: **I am creating packages for later pickup**

#### **Patient 3**

#### *Patient 3 Details*

- Folder Number: DD003
- Patient Details: Daisy Duck, 01 Jan 1980, Female
- Episode: New Patient on 01 Jan 2010 at **Main Clinic**

#### *Prescription 3 Details*

- Capture Date 01 Jan 2010
- Doctor: John Smith
- Duration: 1 month
- Drugs: 3TC 150 - D4T 30 - NVP 200 (1B-30)

#### *Dispensing 3.1 Details*

- Date Packed: 01 Jan 2010
- Package Contains: 1 month supply
- Dispensing Mode: **Dispense Directly**

#### *Dispensing 3.2 Details*

- Date Packed: 15 Jan 2010
- Package Contains: 1 month supply
- Dispensing Mode: **I am creating packages for later pickup**

#### **Patient 4**

#### *Patient 4 Details*

- Folder Number: DD004
- Patient Details: Duey Duck, 01 Jan 1980, Male
- Episode: New Patient on 01 Jan 2010 at **Main Clinic**

#### *Prescription 4 Details*

- Capture Date 01 Jan 2010
- Doctor: John Smith
- Duration: 3 months
- Drugs: 3TC 150 - D4T 30 - EFV 600 (1A-30)

#### *Dispensing 4.1 Details*

- Date Packed: 01 Jan 2010
- Package Contains: 1 month supply
- Dispensing Mode: **Dispense Directly**

#### *Down Referral Info*

- Down Referral Clinic: Clinic A
- Date: 15 Jan 2010
- First Time Down Referred

#### *Dispensing 4.2 Details*

- Date Packed: 15 Jan 2010
- Package Contains: 1 month supply
- Dispensing Mode: **I am creating packages for later pickup**

## **TC93X - Test Data for Package Movement Reports (downReferralModel = online - NOT DONE)**

THIS HAS NOT BEEN CHECKED - NOT IMPORTANT FOR CAPRISA

#### **Test Data**

Set the idart property **downReferralModel** to **online**

## **Patient 1**

### *Patient 1 Details*

- Folder Number: DD001
- Patient Details: Donald Duck, 01 Jan 1980, Male
- Episode: New Patient on 01 Jan 2010 at **Main Clinic**

### *Prescription 1 Details*

- Capture Date 01 Jan 2010
- Doctor: John Smith
- Duration: 1 month
- Drugs: 3TC 150 - D4T 30 - EFV 600 (1A-30)

### *Dispensing 1 Details*

- Date Packed: 01 Jan 2010
- Package Contains: 1 month supply
- Dispensing Mode: **I am creating packages for later pickup**

## **Patient 2**

### *Patient 2 Details*

- Folder Number: DD002
- Patient Details: Daffy Duck, 01 Jan 1980, Female
- Episode: Transferred In on 01 Jan 2010 at **Clinic A**

### *Prescription 2 Details*

- Capture Date 01 Jan 2010
- Doctor: John Smith
- Duration: 1 month
- Drugs: 3TC 150 - D4T 30 - NVP 200 (1B-30)

### *Dispensing 2 Details*

- Date Packed: 01 Jan 2010
- Package Contains: 1 month supply
- Dispensing Mode: **I am creating packages for later pickup**

## **Patient 3**

### *Patient 3 Details*

- Folder Number: DD003
- Patient Details: Daisy Duck, 01 Jan 1980, Female
- Episode: New Patient on 01 Jan 2010 at **Main Clinic**

### *Prescription 3 Details*

- Capture Date 01 Jan 2010
- Doctor: John Smith
- Duration: 1 month
- Drugs: 3TC 150 - D4T 30 - NVP 200 (1B-30)

### *Dispensing 3.1 Details*

- Date Packed: 01 Jan 2010
- Package Contains: 1 month supply
- Dispensing Mode: **Dispense Directly**

### *Dispensing 3.2 Details*

- Date Packed: 15 Jan 2010
- Package Contains: 1 month supply

- Dispensing Mode: **I am creating packages for later pickup**

#### **Patient 4**

##### *Patient 4 Details*

- Folder Number: DD004
- Patient Details: Duey Duck, 01 Jan 1980, Male
- Episode: New Patient on 01 Jan 2010 at **Main Clinic**

##### *Prescription 4 Details*

- Capture Date 01 Jan 2010
- Doctor: John Smith
- Duration: 3 months
- Drugs: 3TC 150 - D4T 30 - EFV 600 (1A-30)

##### *Dispensing 4.1 Details*

- Date Packed: 01 Jan 2010
- Package Contains: 1 month supply
- Dispensing Mode: **Dispense Directly**

##### *Down Referral Info*

- Down Referral Clinic: Clinic A
- Date: 15 Jan 2010
- First Time Down Referred

##### *Dispensing 4.2 Details*

- Date Packed: 15 Jan 2010
- Package Contains: 1 month supply
- Dispensing Mode: **I am creating packages for later pickup**

## **TC911 - Patient History Report**

### **Purpose**

Check that the user can access an accurate **Patient History Report**

### **Prerequisites**

1. Load the test database
2. Add a new patient
3. Capture a prescription
4. Dispense the 1st issue of the prescription
5. Dispense the 2nd issue of the prescription
6. Capture a new prescription
7. Dispense the 1st issue of the new prescription

### **Test Data**

#### *Patient Details*

- Folder Number: JS001
- Patient Details: Joe Soap, Male, 01 Jan 1980
- New Patient episode on 01 Jan 2010 at Main Clinic

#### *1st Prescription Details*

- Capture Date 01 Jan 2010
- Doctor: Test Doctor
- Duration: 3 months
- Drugs: 3TC 150 - D4T 30 - EFV 600

#### **1st Dispensing Details**

- Date Packed: 01 Jan 2010
- Package Contains: 1 month supply
- Dispensing Mode: I am dispensing directly to patients
- After clicking the 'Create Package', say 'Yes' to ARV Start Date of 01 Jan 2010

#### **2nd Dispensing Details**

- Date Packed: 29 Jan 2010
- Package Contains: 2 months supply
- Dispensing Mode: I am dispensing directly to patients
- Pill Count: 4, 4, 2 for 3TC, D4T, EFV (respectively)

#### **2nd Prescription Details**

- Capture Date: 26 Mar 2010
- Doctor: Test Doctor
- Duration: 1 month
- Drugs: 3TC 150 - AZT 300 - EFV 600

#### **3rd Dispensing Details**

- Date Packed: 26 Mar 2010
- Package Contains: 1 months supply
- Dispensing Mode: I am dispensing directly to patients
- Pill Count: 14, 14, 7 for 3TC, D4T, EFV (respectively)

#### **Steps**

1. Log in as admin
2. Click on the **Reports** button
3. Select the **Patient History Report**
4. Type in **JS001** and press **Enter**
5. Check the report against the example attached to this test case.

#### **Notes & Questions**

1. Close the patient's current episode and check that the report shows this info.
2. Change the patient's folder number and check that the report shows this info.

## **TC912 - Episodes Started Report**

#### **Purpose**

Check that the user can access an accurate **Episodes Started Report**

#### **Prerequisites**

1. Load the test database
2. Add the following drug groups to the test database
  - a. 3TC150 - D4T30 - EFV600 as Regimen 1
  - b. AZT300 - DDI400 - LPV/r 133/3 as Regimen 2
3. Add six patients and dispense to some (see below for test data)

#### **Test Data**

##### **Patient 1 Details (not shown in 'started', shown in 'ended' with no drugs)**

- Folder Number: 001
- Patient Details: Joe Soap, Male, 01 Jan 1980
- Episodes:
  - Start: New Patient on 01 Jan 2010

- End: Transferred Out on 15 Mar 2010
- Prescription: None
- Dispensing: None

**Patient 2 Details (shown in 'started' with drugs, shown in 'ended' with drugs)**

- Folder Number: 002
- Patient Details: Jane Soap, Female, 01 Jan 1980
- Episodes:
  - Start: Transferred In on 15 Mar 2010
  - End: Transferred Out on 20 Mar 2010
- Prescription:
  - Capture Date: 15 Mar 2010
  - Drugs: 3TC150 - D4T30 - EFV600 (1A)
- Dispensing:
  - Pack Date: 15 Mar 2010
  - Directly
  - 1 months supply

**Patient 3 Details (shown in 'started' with no drugs, not shown in 'ended')**

- Folder Number: 003
- Patient Details: Jane Soap, Female, 01 Jan 1980
- Episodes:
  - Start: Unknown on 20 Mar 2010
- Prescription:
  - Capture Date: 20 Mar 2010
  - Drugs: 3TC150 - D4T30 - EFV600 (1A)
- Dispensing: None

**Patient 4 Details (shown in 'started' with no drugs, shown in 'ended' with no drugs)**

- Folder Number: 004
- Patient Details: Joe Soap, Male, 01 Jan 1980
- Episodes:
  - Start: Visitor In on 20 Mar 2010
  - End: Visitor Out on 25 Mar 2010
- Prescription:
  - Capture Date: 20 Mar 2010
  - Drugs: AZT300 - DDI400 - LPV/r 133/3 (Regimen 2)
- Dispensing:
  - None

**Patient 5 Details (shown in 'started' with no drugs, not shown in 'ended')**

- Folder Number: 005
- Patient Details: Jane Soap, Unknown, 01 Jan 1980
- Episodes:
  - Start: Restart ART on 18 Mar 2010
- Prescription:
  - Capture Date: 20 Mar 2010
  - Drugs: AZT300 - DDI400 - LPV/r 133/3 (Regimen 2)
- Dispensing
  - Date Packed: 20 Mar 2010
  - Later Pickup
  - 1 months supply
  - Don't scan out to the patient

**Patient 6 Details**

**(shown in 'started' with no drugs, not shown in 'ended')**

- Folder Number: 006
- Patient Details: Jane Soap, Unknown, 01 Jan 1980
- Episodes:
  - Start: Start PEP on 17 Mar 2010
  - End: Stop PEP on 29 Mar 2010
- Prescription:
  - Capture Date: 17 Mar 2010
  - Drugs: AZT300 - DDI400 - LPV/r 133/3 (Regimen 2)
- Dispensing
  - Date Packed: 17 Mar 2010
  - Later Pickup
  - 1 months supply
  - Scan to Patient: 25 Mar 2010

## Steps

1. Log in as admin
2. Click on the **Reports** button
3. Select the **Episodes Started or Ended Report**
4. Select 'Main Clinic' and select 'Started during period'
5. Select 'Patient ID' and 'Patient Name' as the ordering options.
6. Select Start Date of '01 March 2010' and End Date of '31 March 2010'
7. Check the report against the example attached to this test case.

## Notes & Questions

1. Change the **ordering options** and check the report for this ordering.

## TC913 - Episodes Ended Report

Episodes Ended Report

## TC914 - Package Tracking Report

### Purpose

Check that the user can access an accurate **Package Tracking Report**

### Prerequisites

1. Load the test database
2. Set the idart property **downReferralMode** to *offline*
3. Add a new clinic called 'Clinic A'
4. Add a new patient
5. Capture a prescription
6. Dispense the 1st issue of the prescription
7. Down Refer the Patient to 'Clinic A'
8. Dispense the 2nd issue of the prescription for later pickup
9. Scan out package to patient
10. Dispense the 3rd issue of the prescription for later pickup
11. Return Uncollected Package to the pharmacy

### Test Data

#### **Patient Details**

- Folder Number: MM001
- Patient Details: Mickey Mouse, Male, 01 Jan 1980
- Episode: New Patient on 01 Jan 2010 at Main Clinic

#### **1st Prescription Details**

- Capture Date 01 Jan 2010

- Doctor: John Smith
- Duration: 1 month
- Drugs: 3TC 150 - D4T 30 - EFV 600 (1A-30)

#### **1st Dispensing Details**

- Date Packed: 01 Jan 2010
- Package Contains: 1 month supply
- Dispensing Mode: I am dispensing directly to patients
- After clicking the 'Create Package', say 'Yes' to ARV Start Date of 01 Jan 2010

#### **Down Referral Details**

- Down Referral Clinic: Clinic A
- Date Down Referred: 01 Jan 2010
- First Time at Down Referral Clinic: Yes

#### **2nd Dispensing Details**

- Date Packed: 15 Jan 2010
- Package Contains: 1 months supply
- Dispensing Mode: I am creating packages for later pickup

#### **DownReferralMode = Offline: Scan Out Package to Patient Details**

- Collections At: Clinic A / Date Collected: 29 Jan 2010 / Pill Count: 4, 4, 2 for D4T, 3TC and EFV respectively

#### **DownReferralMode = Offline: Scan Package Along (see 'Notes & Questions' below)**

- Scan Out Package to Down Referral Clinic:
  - Clinic A / Date Left: 16 Jan 2010
- Log out of pharmacy and log in to Clinic A.
- Packages Arrive:
  - Clinic A / Date Arrived: 17 Jan 2010
- Patient Collects:
  - Clinic A / Date Collects: 29 Jan 2010 / Pill Count: 4, 4, 2 for D4T, 3TC and EFV respectively

#### **3rd Dispensing Details**

- Date Packed: 12 Feb 2010
- Package Contains: 1 months supply
- Dispensing Mode: I am creating packages for later pickup

#### **Return Uncollected Package to Pharmacy Details**

- Reason for Return: No longer receiving treatment at clinic
- Date of Return: 26 Feb 2010
- What to do with drugs: Return drugs to stock (can be redispensed to other patients)
- Episode Stop Reason: Stopped ART by Clinician
- Episode Stop Date: 26 Feb 2010

#### **Steps**

1. Log in as admin
2. Click on the **Reports** button
3. Select the **Package Tracking Report**
4. Type in **MM001** and press **Enter**
5. Check the report against the example attached to this test case.

#### **Notes & Questions**

1. Change the idart.property **downReferralMode** to **online** and repeat with a 2nd patient (MM002) with all other data fields the same as the patient above (besides the scanning along). Check it against the attached report.

# TC921 - Monthly Receipts and Issues Report

## Purpose

Check that the user is able to generate a Monthly issues and receipts and issues report for each pharmacy correctly\*

## Prerequisites

1. Load the test database
2. Add the following drug groups to the test database
  - a. 3TC150 - D4T30 - NVP200 as Regimen 1
  - b. AZT300 - DDI400 - LPV/r 133/3 as Regimen 2
3. Add another pharmacy called Paed Drugs

## Test Data

- Add 10 units of 3TC150 - D4T30 - EFV600

### **3TC 15mg (Same details need to be used for D4T and NVP)**

- Manufacturer: GSK
- Batch Number: abc123
- Expiry Date: January year(2 years after current date)
- Units : 10
- Pharmacy: main
- Arrival date: Current date

## Steps

1. Log in as admin
2. Click on the **Stock & Dispensing** button
3. Click on the **Stock Arrives at the Pharmacy** button
4. Add 10 units of 3TC 150mg
5. Add 10 units of D4T 30mg
6. Add 10 units of NVP 200mg
7. Click on the Save button
8. Click yes to view Stock Report
9. Click on **Make up Drug Packages for Patients** button and dispense a package for patient 123
10. After the packages have been created navigate to the Reports section
11. Select **Monthly Stock Receipts and Issues Report**
12. Select the applicable month that the report needs to generate.
13. Check the report against the example attached to this test case.

## Notes & Questions

1. Check that stock on hand is correct based on how much stock was received and issued.

# TC926 - ARV Drug Usage

## Purpose

Check that the user is able to generate an ARV Drug Usage Report for each pharmacy correctly\*

## Prerequisites

1. Load the test database
2. Add the following drug groups to the test database
  - a. 3TC150 - D4T30 - NVP200 as Regimen 1
  - b. AZT300 - DDI400 - LPV/r 133/3 as Regimen 2
3. Add another pharmacy called Paed Drugs

## Test Data

- Add 10 units of 3TC150 - D4T30 - EFV600

#### **3TC 15mg (Same details need to be used for D4T and NVP)**

- Manufacturer: GSK
- Batch Number: abc123
- Expiry Date: January year(2 years after current date)
- Units : 10
- Pharmacy: main
- Arrival date: Current date

#### **Steps**

1. Log in as admin
2. Click on the **Stock & Dispensing** button
3. Click on the **Stock Arrives at the Pharmacy** button
4. Add 10 units of 3TC 150mg
5. Add 10 units of D4T 30mg
6. Add 10 units of NVP 200mg
7. Click on the Save button
8. Click yes to view Stock Report
9. Click on **Make up Drug Packages for Patients** button and dispense a package for patient 123
10. After the packages have been created navigate to the Reports section
11. Select **ARV drug usage Report**
12. Select the drugs that were dispensed to patient 123 and click View report.
13. Check the report against the example attached to this test case.

#### **Notes & Questions**

## **TC931 - Packages Created**

#### **Purpose**

Check that the user can access an accurate **Packages Created**

#### **Prerequisites**

1. Load the test database
2. Set the iDART property **downReferralMode** to **offline**
3. Add a new clinic called 'Clinic A'
4. Add Patient 1, capture a prescription and dispense to them (for later pickup)
5. Add Patient 2, capture a prescription and dispense to them (for later pickup)
6. Add Patient 3, capture a prescription, dispense 1st issue directly, 2nd issue for later pickup
7. Add Patient 4, capture a prescription, dispense 1st issue directly, down refer the patient, create 2nd issue for later pickup at Clinic A

#### **Test Data**

See [Test Data for Package Movement Reports \(downReferralMode = offline\)](#)

#### **Steps**

1. Log in as admin
2. Click on the **Reports** button
3. Select the **Packages Created**
4. Generate the report for the time period: 01 Jan 2010 to 31 Jan 2010 for both **Main Clinic** and **Clinic A**
5. Check the two reports against the attached PDF.

#### **Notes & Questions**

1. Change the idart property **downReferralMode** to **online** and retest with the same data. Compare against the respective attached PDF reports.

## TC932 - Packages Leaving Pharmacy

### Purpose

Check that the user can access an accurate **Packages Leaving Pharmacy**

### Prerequisites

1. Load the test database
2. Set the iDART property **downReferralMode** to **offline**
3. Add a new clinic called 'Clinic A'
4. Add Patient 1, capture a prescription and dispense to them (for later pickup)
5. Add Patient 2, capture a prescription and dispense to them (for later pickup)
6. Add Patient 3, capture a prescription, dispense 1st issue directly, 2nd issue for later pickup
7. Add Patient 4, capture a prescription, dispense 1st issue directly, down refer the patient, create 2nd issue for later pickup at Clinic A

### Test Data

See [Test Data for Package Movement Reports \(downRefferalMode = offline\)](#)

### Steps

1. Log in as admin
2. Click on the **Reports** button
3. Select the **Packages Leaving Pharmacy**
4. Generate the report for the time period: 01 Jan 2010 to 31 Jan 2010 for both **Main Clinic** and **Clinic A**
5. Check the two reports against the attached PDF.

### Notes & Questions

1. Change the idart property **downReferralMode** to **online** and retest with the same data. Compare against the respective attached PDF reports.

## TC933 - Packages Received at Clinic

### Purpose

Check that the user can access an accurate **Packages Received at Clinic**

### Prerequisites

1. Load the test database
2. Set the iDART property **downReferralMode** to **offline**
3. Add a new clinic called 'Clinic A'
4. Add Patient 1, capture a prescription and dispense to them (for later pickup)
5. Add Patient 2, capture a prescription and dispense to them (for later pickup)
6. Add Patient 3, capture a prescription, dispense 1st issue directly, 2nd issue for later pickup
7. Add Patient 4, capture a prescription, dispense 1st issue directly, down refer the patient, create 2nd issue for later pickup at Clinic A

### Test Data

See [Test Data for Package Movement Reports \(downReferralMode = offline\)](#)

### Steps

1. Log in as admin
2. Click on the **Reports** button
3. Select the **Packages Received at Clinic**
4. Generate the report for the time period: 01 Jan 2010 to 31 Jan 2010 for both **Main Clinic** and **Clinic A**
5. Check the two reports against the attached PDF.

### Notes & Questions

1. Change the idart property **downReferralMode** to online and retest with the same data. Compare against the respective attached PDF reports.

## TC934 - Packages Collected

### Purpose

Check that the user can access an accurate **Packages Collected**

### Prerequisites

1. Load the test database
2. Set the iDART property **downReferralMode** to **offline**
3. Add a new clinic called 'Clinic A'
4. Add Patient 1, capture a prescription and dispense to them (for later pickup)
5. Add Patient 2, capture a prescription and dispense to them (for later pickup)
6. Add Patient 3, capture a prescription, dispense 1st issue directly, 2nd issue for later pickup
7. Add Patient 4, capture a prescription, dispense 1st issue directly, down refer the patient, create 2nd issue for later pickup at Clinic A

### Test Data

See [Test Data for Package Movement Reports \(downReferralMode = offline\)](#)

### Steps

1. Log in as admin
2. Click on the **Reports** button
3. Select the **Packages Collected**
4. Generate the report for the time period: 01 Jan 2010 to 31 Jan 2010 for both **Main Clinic** and **Clinic A**
5. Check the two reports against the attached PDF.

### Notes & Questions

1. Change the idart property **downReferralMode** to online and retest with the same data. Compare against the respective attached PDF reports.

## TC935 - Packages Awaiting Pickup

### Purpose

Check that the user can access an accurate **Packages Awaiting Pickup Report**

### Prerequisites

1. Load the test database
2. Set the iDART property **downReferralMode** to **offline**
3. Add a new clinic called 'Clinic A'
4. Add Patient 1, capture a prescription and dispense to them (for later pickup)
5. Add Patient 2, capture a prescription and dispense to them (for later pickup)
6. Add Patient 3, capture a prescription, dispense 1st issue directly, 2nd issue for later pickup
7. Add Patient 4, capture a prescription, dispense 1st issue directly, down refer the patient, create 2nd issue for later pickup at Clinic A

### Test Data

See [Test Data for Package Movement Reports \(downReferralMode = offline\)](#)

### Steps

1. Log in as admin
2. Click on the **Reports** button
3. Select the **Packages Awaiting Pickup Report**
4. Check that Patient 3 (DD003) and only this patient is shown in the table when **Main Clinic** is selected in the drop down menu.
5. Click 'View Report' for Main Clinic and check the report against the attached PDF.

- Check that Patient 2 (DD002) and Patient 4 (DD004) is shown in the table when **Clinic A** is selected in the drop down menu.
- Click 'View Report' for Clinic A and check the report against the attached PDF.

#### Notes & Questions

- Change the idart property **downReferralMode** to online and retest with the same data. Compare against the respective attached PDF reports.

## TC936 - Patients Expected On A Day

### Purpose

Check that the user can access an accurate **Patients Expected on A Day Report**

### Prerequisites

- Load a current client's database (minimum 1 week old)
- Run the *RemovePatientNames.sql* script (available on albert/sensitive)
- Add a new patient (**Main Clinic**), capture a prescription & dispense directly 1 month supply
- Add a new patient (**Down Referral Clinic**), capture a prescription & dispense directly 2 months supply
- Add a new patient (**Main Clinic**), capture a prescription & dispense 1 month supply directly, **down refer**, dispense for **later pickup**

### Test Data

#### *Patient 1 Details*

- TEST1 / Test1 Patient1 / 01 Jan 1980 / Male
- Episode: New Patient on TODAY at **Main Clinic**
- Prescription: TODAY, Test Doctor, 3 months, 1A (3TC 150 - D4T 30 - EFV 600)
- Dispensing: TODAY, **1 month supply, directly**

#### *Patient 2 Details*

- TEST2 / Test2 Patient2 / 01 Jan 1980 / Female
- Episode: New Patient on TODAY at a **Down Referral Clinic**
- Prescription: TODAY, Test Doctor, 3 months, 1A (3TC 150 - D4T 30 - EFV 600)
- Dispensing: TODAY, **2 months supply, directly**

#### *Patient 3 Details*

- TEST3 / Test3 Patient3 / 01 Jan 1980 / Female
- Episode: New Patient on TODAY at **Main Clinic**
- Prescription: TODAY, Test Doctor, 3 months, 1A (3TC 150 - D4T 30 - EFV 600)
- Dispensing: TODAY, **1 months supply**, directly
- Down Refer:** TODAY, to **Down Referral Clinic**, First Time
- Dispensing: TODAY, 1 months supply, for later pickup

### Steps

- Log in as admin
- Click on the **Reports** button
- Select the **Patients Expected on a Day Report**
- 4 weeks from today:
  - Main Clinic** and press 'View Report' - **TEST1** must be shown
  - Down Referral Clinic** and press 'View Report' - **TEST3** must be shown
- 8 weeks from today:
  - Main Clinic** and press 'View Report' - None of the test patients should be shown
  - Down Referral Clinic** and press 'View Report' - **TEST2** must be shown

#### Notes & Questions

- Load a patient who is expected today (and hasn't come yet) and dispense (patient comes on their appointment date) - load report for today + 4 weeks from today - patient should be in both.

- Load a patient who is expected in 2 days time (and hasn't come yet) and dispense (patient comes early)
- Load a patient who was meant to arrive at the clinic 2 days ago (and hasn't come yet) and dispense (patient comes late)

## TC937 - Missed Appointments Report

### Purpose

Check that the user can access an accurate **Missed Appointments Report**

### Prerequisites

- Load the test database
- Add a new patient (**Main Clinic**), capture a prescription & dispense directly 1 month supply. Generate report for TODAY.
- Add a new patient (**Down Referral Clinic**), capture a prescription & dispense directly 2 months supply
- Add a new patient (**Main Clinic**), capture a prescription & dispense 1 month supply directly, **down refer**, dispense for **later pickup**

### Test Data

#### **Patient 1 (patient comes on expected date)**

- TEST1 / Test1 Patient1 / 01 Jan 1980 / Male
- Episode: New Patient on 4 WEEKS AGO at **Main Clinic**
- Prescription: 4 WEEKS AGO, John Smith, 1 month, 1A (3TC 150 - D4T 30 - EFV 600)
- Dispensing: 4 WEEKS AGO, **1 month supply, directly**
- REPORT:** TODAY at Main Clinic, between 0 and 1 days late - TEST1 should be displayed as 0 days late.
- Dispensing: TODAY\*, 1 month supply\*, **directly**
- REPORT:** TODAY at Main Clinic, between 0 and 1 days late - TEST1 should NOT be displayed

#### **Patient 2 (patient comes 7 days late)**

- TEST2 / Test2 Patient2 / 01 Jan 1980 / Female
- Episode: New Patient on 5 WEEKS AGO at a **Down Referral Clinic**
- Prescription: 5 WEEKS AGO, John Smith, 1 month, 1A (3TC 150 - D4T 30 - EFV 600)
- Dispensing: 5 WEEKS AGO, **1 month supply, directly**
- REPORT:** TODAY at Down Referral Clinic, Between 0 and 7 days late - TEST2 should be displayed as 7 days late.
- Dispensing: TODAY\*, 1 month supply\*, **directly**
- REPORT:** TODAY at Down Referral Clinic, Between 0 and 7 days late - TEST2 should NOT be displayed

#### **Patient 3 Details (patient comes 7 days early)**

- TEST3 / Test3 Patient3 / 01 Jan 1980 / Female
- Episode: New Patient on 3 WEEKS AGO at **Main Clinic**
- Prescription: 3 WEEKS AGO, Test Doctor, 1 month, 1A (3TC 150 - D4T 30 - EFV 600)
- Dispensing: 3 WEEKS AGO, **1 months supply, directly**
- REPORT:** 2 WEEKS TIME at Main Clinic, Between 3 and 14 days late - TEST3 should be displayed as 7 days late.
- Dispensing: TODAY, **1 months supply, directly**
- REPORT:** 2 WEEKS TIME at Main Clinic, Between 3 and 14 days late - TEST3 should NOT be displayed

### Steps

- Log in as admin
- Load TEST1 and generate 2 reports
- Load TEST2 and generate 2 reports
- Load TEST3 and generate 2 reports

### Notes & Questions

- None

## TC941 - Drug Combinations (CSV)

### Purpose

Check that the user is able to generate a Drug Combinations Report for all patients

#### Prerequisites

1. Load the test database
2. Add the following drug groups to the test database
  - a. 3TC150 - D4T30 - NVP200 as Regimen 1
  - b. AZT300 - DDI400 - LPV/r 133/3 as Regimen 2
3. at least 2 patients have been created with the above prescriptions

#### Test Data

- Use Patients that have already been created in the test database ie 123 ; 123456789

#### Steps

1. Log in as admin
2. Confirm that patientID (123 & 123456789) have valid prescriptions by checking them in the **Make up Drug Packages for Patients** screen
3. Close the screen and navigate to the Reports section
4. Select the Drug Combinations report found in \*M & E Reports
5. Choose the main clinic and select the current date to run the report
6. Save the CSV file to location that can easily be found.
7. Navigate to the Saved CSV file and open using spreadsheet application
8. Check the report against the example attached to this test case.

#### Notes & Questions

## TC944 - Prescribing Doctors Report

#### Purpose

Check that the user is able to generate a Prescribing doctors report

#### Prerequisites

1. Load the test database
2. Add the following drug groups to the test database
  - a. 3TC150 - D4T30 - EFV600 as Regimen 1
3. Create a new patient and add prescription

#### Test Data

- Add 10 units of 3TC150 - D4T30 - EFV600

#### **3TC 15mg (Same details need to be used for D4T and NVP)**

- Manufacturer: GSK
- Batch Number: abc123
- Expiry Date: January year(2 years after current date)
- Units : 10
- Pharmacy: main
- Arrival date: Current date

#### **Prescribing doctor**

- Name : John
- Surname: Smith

#### **Patient details**

- Patient ID : JT001
- Name :John

- Surname: Smith (co-incidence that it is the same as the doctor)
- DOB: 12 august 1990
- Episode : new patient

#### **Prescription Details**

- regimen 3TC150 - D4T30 - EFV600

#### **Steps**

1. Log in as admin
2. Click on the **Patient Admin**
3. Click on the **Add new patient**
4. Add patient details using data supplied above
5. Click on Create Initial Prescription
6. Add/Select Doctor based on Data provide above
7. Add drug that contains 3TC150 - D4T30 - EFV600
8. Click Save prescription
9. Navigate to the reports section and select the Prescribing Doctors Report
10. Select the period to report
11. Check the report against the example attached to this test case.

#### **Notes & Questions**

## **TC945 - PEPFAR Report**

#### **Purpose**

Check that the user can access an accurate **PEPFAR Report**

#### **Prerequisites**

1. Load the test database
2. Remove the four patients (1, 123456789, ABC123 & 123) using the remove\_test\_patient.sql script (in the 'scripts' folder in the install folder)
3. Add 'Clinic A' as a down referral clinic

#### **Test Data**

##### **Patient 1 (new patient)**

- Patient: TEST1 / Test1 Patient1 / 01 Jan 1980 / Male / New Patient on 01 Jan 2010 at **Main Clinic**
- Prescription: 01 Jan 2010 / John Smith / 1 month / 1A-30 (3TC 150 - D4T 30 - EFV 600)
- Dispensing: 01 Jan 2010 / 1 month supply / directly
- **Later in the Test Case** - Close Episode: Transferred Out / 15 Jan 2010

##### **Patient 2 (transferred in)**

- Patient: TEST2 / Test2 Patient2 / 01 Jan 1980 / Female / Transferred In on 01 Jan 2010 at **Main Clinic** / Pregnant - LMP 1 Dec 2009
- Prescription: 01 Jan 2010 / John Smith / 1 month / 1B-30 (3TC 150 - D4T 30 - NVP 200)
- Dispensing: 01 Jan 2010 / 1 month supply / directly
- **Later in the Test Case** - Close Episode: Stopped ART by Clinician / 15 Jan 2010

##### **Patient 3 (unknown gender)**

- Patient: TEST3 / Test3 Patient3 / 01 Jan 1980 / Unknown / New Patient on 01 Dec 2009 at **Main Clinic**
- Prescription: 01 Dec 2009 / John Smith / 1 month / 1A-30 (3TC 150 - D4T 30 - EFV 600)
- Dispensing: 01 Dec 2009 / 1 month supply / directly
- **Later in the Test Case** - Close Episode: Lost to Follow Up / 1 Feb 2010

##### **Patient 4 (paed)**

- Patient: TEST4 / Test4 Patient4 / **01 Jan 2009** / Female / Transferred In on 01 Jan 2010 at **Main Clinic**
- Prescription: 01 Jan 2010 / John Smith / 1 month / Weight 4.5kg / 3TC 10 (3ml twice) - d4T 1 (6ml twice) - AZT 10 (6ml twice)
- Dispensing: 01 Jan 2010 / 1 month supply / directly
- **Later in the Test Case** - Close Episode: Deceased / 31 Jan 2010

#### **PEPFAR Report**

- 1 Jan 2010 - 31 Jan @ Main Clinic
- Default ages

#### **Steps**

1. Log in as admin
2. Add **TEST1**, capture prescription & dispense. Generate PEPFAR Report.
  - a. Patient should be listed in the **2nd, 4th, 5th & 6th columns**, under **Males, 14+ years, TOTAL MALES** and **TOTAL PATIENTS**
  - b. Patient's prescription should be listed in the **Regimen** table
3. Add **TEST2**, capture prescription & dispense. Generate PEPFAR Report.
4. Generate PEPFAR report ( 1 Jan 2010 - 31 Jan / Main Clinic ).  
  - a. Patient should be listed in the **3rd, 4th, 5th & 6th columns**, under **Females, 14+ years, TOTAL FEMALES, TOTAL PATIENTS** and **Pregnant Females**
  - b. Patient's prescription should be listed in the **Regimen** table
5. Add **TEST3**, capture prescription & dispense. Generate PEPFAR Report.
6. Generate PEPFAR report ( 1 Jan 2010 - 31 Jan / Main Clinic ).  
  - a. Patient should be listed in the **1st, 5th & 6th columns**, under **UNKNOWN** and **TOTAL PATIENTS**
  - b. Patient's prescription should be listed in the **Regimen** table
7. Add **TEST4**, capture prescription & dispense. Generate PEPFAR Report.  
  - a. Patient should be listed in the under **Females, 0-14 years, Females, 0-5 years, TOTAL FEMALES** and **TOTAL PATIENTS**
  - b. Patient's prescription should be listed in the **Regimen** table
8. **Close episodes** for all patients. Episode: Transferred Out / 15 Jan 2010
  - a. Generate PEPFAR Report.
  - b. ???

#### **Notes & Questions**

1. None

## **TC946 - Clinic Indicators Report**

#### **Purpose**

Check that the user can access an accurate **Clinic Indicator Report**

#### **Prerequisites**

1. Load the database with test data.

#### **Test Data**

##### **Patient 1: TEST1**

- **Patient:** TEST1 / Test1 Patient1 / 01 Jan 1980 / Male / New Patient on 01 Jan 2010 at **Clinic A**
- **Prescription:** 01 Jan 2010 / John Smith / 1 month / 1A-30
- **Dispensing:** 01 Jan 2010 / 1 month supply / dispense directly

#### **Steps**

1. Log in as admin
2. Click on the **Reports** button
3. Select the **Clinic Indicator Report**: 22 Mar 2006 - today's date for Main Clinic. Check against attached.
4. Add a new clinic **Clinic A**.
5. Add **TEST1** dfd

- 6.
7. Check the report against the example attached to this test case.

### **Notes & Questions**

1. Close the patient's current episode and check that the report shows this info.
2. Change the patient's folder number and check that the report shows this info.

From a clean database do the following

- Create a patient
- Create a prescription for Patient
- Save and run CIR (clinic indicator report)
  - result should be one patient currently on treatment based on current prescription
  - result should be one patient episode created 'new patient'
  - \*Dispense the package to above patient and run CIR
    - result there should be 1 patient in Total patients initiated during period
- Create another patient with episode transferred in
- create a prescription for transferred patient
- Dispense to transferred patient
- run CIR report
  - result should be that there are 2 patients currently on treatment
  - result should be that total episodes started should be 2 with episode started detail should have a patient transferred in and new patient
- Stop episode of new patient with reason 'other'
- Down refer transferred in patient to down referal clinic created
  - run CIR report
    - result should be that 0 patients currently on treatment
    - result should be 2 episodes stopped with details down referred and 'other'
- run CIR for down referal clinic
  - result should be 1 patient currently on treatment
  - result should be 1 episode started reason down referred
  - check that there are 0 episodes closed

P1: new patient + prescription + dispense

P2: transfer in + prescription + dispense

then stop:

P1: Transfer Out

P2: Down Referred

Notes:

- Number of Adult Patients on Treatment (based on current prescription) at 31 Jan 2010
- Number of Adult Patients ever initiated at on treatment (based on packages received)
- Check for paeds
- Episodes started and ended before period
- Episode started before period and ended during period
- Episode started before period and episode hasn't closed
- Episode started in this period and ended after this period

### **TC947 - Clinic Indicator Report (Draft)**

P1: new patient + prescription + dispense

P2: transfer in + prescription + dispense

then stop:

P1: Transfer Out

P2: Down Referred

## Notes:

- Number of Adult Patients on Treatment (based on current prescription) at 31 Jan 2010
- Number of Adult Patients ever initiated at on treatment (based on packages received)
- Check for paeds
- Episodes started and ended before period
- Episode started before period and ended during period
- Episode started before period and episode hasn't closed
- Episode started in this period and ended after this period

## Test Cases to be completed

- Adding/Saving patients
- creating prescriptions
- dispensing directly
- dispensing for later pickup to main clinic patient
- dispensing for later pickup to down referred patients
- scan package to patient (main clinic)
- scan package out to clinic
- scan package to down referred patient
- Dispens 3 months supply directly
- dispense 3 months supply for later pickup
- Update/Change patient prescription
- Enter pill counts for returning patients
- Change appointments in update an existing patient details
- Add a new patient to the main clinic
- Add a patient to a down referred clinic
- Check Save function from Add patient screen to prescription screen
- Check All reports
  - clinic indicator report after each patient has been created
  - check the clinic indicator report to confirm that there is one episode closed and the numbers are correct
  - Check the episodes started and ended report
  - check the patient history report for each patient that has been created after they have been dispensed to
  - Check all package related reports for the down referred patient
- When scanning patients make sure the correct patient is displayed ( scan another patient immediately afterwards to confirm)
- Merge 2 patients belonging to the same clinic
- Create 2 down referral clinics and make sure that the main clinic is defaulted at the login
- Create two new users making sure that they automatically have access to the previously created down referral clinics
- create 2 doctors and make one inactive to check if they are displayed in the list and vice versa for the other doctor

## Updating from iDART 3.0 to latest

Edit the 'expected database version' in the 'Simple Domain Table' from 3.0 to 3.1 as the update script does not run correctly for that version.

After that has been done the implementer can now continue running the update scripts.

## Notes



### Temporary Notes

This is a space for temporary notes. These notes should be removed when no-longer needed.

## Update to down referral modes

- Change the 'IgnoreScanInAtClinic' variable to be modal - with settings of 'comprehensiveOnline', 'comprehensiveOffline' and 'easyOffline' (names of course can change)
- Rearrange Stock & Dispensing Screen based on this variable
- Dispensing Screen: set 1st 2 dates of package if easyOffline when selecting 'create package for later pickup' and set 2nd 2 dates when scan out to patient
- Change Scan Package to Patient Screen: Enable the drop down list of clinics, change query to show all packages with clinics with 2 dates, mainclinic with 1 date (although possibly change this in dispensing screen?). Set last 2 dates if easyOffline
- Update Documentation (in Support Manual - regarding variable AND in User Manual - down referral)
- Modify Packages Awaiting Pickup Report to include barcodes (I'm not sure if this was the final agreed report to modify though, vaguely remembered saying we should change the Packages Created one?) Use "packages leaving pharmacy report" to show packages?
- Use patient id as well as package id in scan to patient screen. What happens when there is more than one package for the patient?

## Notes on creating a "CIDA" type connection to Communicate

NOTE: The code we worked off for Vicky's Vodacom project is <http://svn.cell-life.org/svn/idart/idart/branches/iDART-3.8.1.1>

1. Create a new user on communicate. This user must belong to the organisation in question.
2. Give the user the role of "Organisation Admin" (on Communicate).
3. Create both a FLEXI and a DAILY campaign on Communicate, **and activate them**.  
*NB: It is very important to activate the campaigns or they will not show up in iDart.*
4. Add the user detail and the URL to the communicate server to the sms.properties file in iDart.

```
# Communicate login details
communicateUrl = http://communicate.cell-life.org/
communicateUsername = idarttest
communicatePassword = password
```

5. In the idart.properties file, set cidaStudy=true.

```
cidaStudy=true
```

6. In iDart, there is a dropdown menu where you select what campaign you want to add patients to.

## Appointment Reminders

The following reminders can be set in the properties file:

```
appointmentReminder1_daysBefore = 1
appointmentReminder2_daysBefore = 1
missedAppointmentSms1_daysLate = 1
missedAppointmentSms2_daysLate = 3
missedAppointmentSms3_daysLate = 7
missedAppointmentSms4_daysLate = 14
```

- Appointment reminders are sent from iDart, so if iDart is not open or if the computer is not online, **they will not be sent**.
- These reminders are sent via a *Just Send SMS* campaign on Communicate.

- They are not sent at any specific time, they are fired off when iDart starts up.
- By setting the value to "-1" (negative 1) it should leave out that particular message.

## **Old wiki archive**

### **checklists**

#### **Checklists**

- Checklists
  - Alpha release checklist
    - Pre release preparation
    - Create release
    - Increment the release version number
    - Increment the trunk version number
    - Create new build on Hudson for new release
    - Example
  - Beta release checklist
    - Pre release preparation
    - Create release
    - Increment the release version number
  - Public release checklist

## **Alpha release checklist**

### ***Pre release preperation***

- Verify all tickets have been completed to BA satisfaction.
- Merge all bug fixes from existing release branches into the trunk.
- Use the [SVN merge](#) wiki page to decide what needs to be merged and update it if any merges are performed.

### ***Create release***

- Create release branch and switch local copy to branch e.g. iDART-3.1
- Create tag for release e.g. iDART-3.1.0-ALPHA
- Update changelog on wiki ([Change log](#))
- Can use revision log to view changes e.g. [-trunk](#)
- Can use tickets for that release milestone.
- Generate installer and remove installer build number (idart-install-3.1.0.jar)
- Generate database scripts (see metadata/database/generateScripts.py)
- python generateScripts.py -h nkosi.cell-life.org -u Cell-Life -v 3.2.0 -V 3.1.0 minimal demo
- Create release pack (idart-3.1.0\_alpha.zip):
- iDART installer (idart-install-3.1.0.jar)
- sqldiff script (idart\_3.0.0\_to-latest-sqldiff.sql)
- demo db script (idart\_3.1.0\_demo.backup)
- minimal db script (idart\_3.1.0\_minimal.backup)
- installation guide (iDARTInstallationGuide.html + idart\_install\_img folder)
- Copy release pack to release folder (albert/Unfiled/iDART  
- Testing/Installations/releases)

### ***Increment the release version number***

- Increment the revision version number in build.properties and commit it to SVN
- use comment like: RELEASE. Start of 3.1.1-alpha

### ***Increment the trunk version number***

- Switch local copy back to trunk (or checkout trunk as new project).
- generateScripts using "fresh" argument instead of "minimal or demo"
- update version numbers in build.properties to 3.4.0
- update readme in metadata, distrib, readme.txt to 3.4.X for all version numbers
- Increment the revision version number in build.properties and commit it to SVN.

### ***Create new build on Hudson for new release***

- Create new build on Hudson for new release

### ***Example***

```
/trunk : idart.version.short = 3.2.0-alpha  
/trunk -> /branches/iDART-3.2  
/branches/iDART-3.2 : create release installer etc. ending with build -> clean  
/branches/iDART-3.2 : idart.version.short -> change from 3.2.0-alpha to 3.2.1-alpha (commit to svn)  
/trunk : idart.version.short -> change from 3.2.0-alpha to 3.3.0-alpha
```

## **Beta release checklist**

### ***Pre release preperation***

- Verify all tickets have been completed to BA satisfaction.
- Merge all bug fixes from existing release branche into the trunk.

- Use the [SVN merge](#) wiki page to decide what needs to be merged and update it if any merges are performed.

### **Create release**

- Create tag for release e.g. iDART-3.1.5-beta
- Update changelog on wiki ([Change log](#))
- Can use revision log to view changes e.g. `-trunk`
- Can use tickets for that release milestone.
- Generate installer and remove installer build number (`idart-install-3.1.5.jar`)
- Generate database scripts (see `metadata/database/generateScripts.py`)
- `python generateScripts.py -h nkosi.cell-life.org -u Cell-Life -v 3.1.5 -V 3.1.0 minimal demo`
- Create release pack (`idart-3.1.5_beta.zip`):
- iDART installer (`idart-install-3.1.5.jar`)
- sqldiff script (`idart_3.1.0_to-latest-sqldiff.sql`)
- demo db script (`idart_3.1.5_demo.backup`)
- minimal db script (`idart_3.1.5_minimal.backup`)
- installation guide (`iDARTInstallationGuide.html` + `idart_install_img` folder)
- Copy release pack to release folder (albert/Unfiled/iDART  
- Testing/Installations/releases)

### **Increment the release version number**

- Increment the revision version number in `build.properties` and commit it to SVN
- use comment like: RELEASE. Start of 3.1.6-beta

### **Public release checklist**

A public release is the same as a Beta release except that the release pack is uploaded to [sourceforge](#).

- Create the release pack e.g. `idart-3.4.0.zip`
- Create the release on SourceForge ([how to create a release on sourceforge](#))
- Create a news item on the website informing people about the release.
- Send an email to the implementers and developers lists about the release.

## **conventions**

### **iDART Conventions**

- iDART Conventions
  - Code structure
  - Database
    - Hibernate
    - Database changes and updates
      - Making changes to the Database
  - Operating system resources (Fonts, Colors, Images)
  - Reports
  - General things to avoid

In order to successfully collaborate, we must all try our best to abide with some common conventions. Failure to follow these conventions will inevitably cause more work for others, so your attention to them is much appreciated.

## Code structure

- GUI framework
- GUI Structure Restructure
- Data Managers
- iDART properties

## Database

### Hibernate

iDART makes use of [Hibernate](#) as the database abstraction layer. Whilst this has many advantages it also means that certain coding disciplines need to be maintained to avoid unexpected behavior:

*Hibernate session management conventions*

*Persistence using JPA, with Hibernate as the provider*

### Database changes and updates

Currently the iDART development team uses a set of scripts to perform database updates. These scripts are as follows:

- `idart_x.x.0_schema.sql`
- Script to drop/generate the iDART database (with NO data)
- `idart_x.x.0_coredata.sql`
- Script to populate tables with REQUIRED data only
- `idart_x.x.0_testdata.sql`
- Script to populate tables with test data
- `idart_x.x.x_to-y.y.y-sqldiff.sql`
- Script to update db version x.x.x to db version y.y.y
- `idart_x.x.x_to-latest-sqldiff.sql`
- Script to update a database to the latest version.

When creating a new database these scripts should be run in the following order:

1. `idart_x.x.0_schema.sql`
2. `idart_x.x.0_coredata.sql`
3. `idart_x.x.0_testdata.sql` (optional)
4. `idart_x.x.x_to-latest-sqldiff.sql`

An existing database can be updated by running the appropriate `idart_x.x.x_to-y.y.y-sqldiff.sql` files (if necessary) followed by the `idart_x.x.x_to-latest-sqldiff.sql` file.

See [Creating database scripts](#) for more info on creating and using the scripts.

### Making changes to the Database

Changes to the database effect all developers and hence before a change is made the person wanting to make the change **MUST** send and email with the proposed change (either in the form of a DB script or a model diagram) to the idart-developers mailing list. Once everybody has had a chance to give feedback and agreement has been reached, the changes should be added to the `idart_x.x.x_to-y.y.y-sqldiff.sql` script and committed to SVN.

## Operating system resources (Fonts, Colors, Images)

Previously, each iDART GUI loaded Fonts, Images and Colors as needed, as in the examples below:

```
#!java
Image image = new Image(Display.getCurrent().getClass().getResourceAsStream("/_images_pharmacy/Report_PatientHistory_30x26.jpg"));
```

```
}}}
```

Whilst there is nothing wrong with this it is a little clumsy and also leaves the developer with the responsibility with disposing of the image (which was never done in iDART).

Fonts (and Colors) are similar:

```
#!java
static Font font_8_VeraSans_italic = new
Font(Display.getDefault(),"Bitstream Vera Sans", 8,SWT.ITALIC);
```

and were also never disposed. For an explanation of why this isn't ideal, see <http://www.eclipse.org/articles/swt-design-2/swt-design-2.html>

This has now changed, and JFace Registries are being used instead. Registries maintain references to resources, and load them as needed. They also handle disposing resources. A resource is stored in a registry with a symbolic name, which can be used to access the resource.

This is done in class org.celllife.idart.gui.utils.ResourceUtils.

To load and use fonts in the FontRegistry, use the following code:

```
#!java
//add the font
fontRegistry.put("font_24_VeraSans", new FontData[] { new
FontData("Bitstream Vera Sans", 24, SWT.NORMAL) });

//use the font
btnPatientAdmin.setFont(ResourceUtils.getFont("font_24_VeraSans"));
```

To load and use colors in the Color Registry, use the following:

```
#!java
//add the color
colorRegistry.put("red",
Display.getDefault().getSystemColor(SWT.COLOR_RED).getRGB());

//use the color
grpTitle.setBackground(ResourceUtils.getColor("red"));
```

To load a new Image into the ImageRegistry, just place it in the img/ directory, which is automatically scanned when iDART loads using the following code:

```
#!java
File imageDirectory = new File("img");
for (File f : imageDirectory.listFiles()) {
    try {
        imageRegistry.put(f.getName(), ImageDescriptor
.createFromURL(new URL((new File(".")).toURI().toURL(),
"img" + File.separator + f.getName())));
    } catch (MalformedURLException m) {
        //TODO handle MalformedURLException
    }
}
```

You can then refer to the image by its (case-sensitive) filename:

```
#! java
lblPicLogoff.setImage(ResourceUtils.getImage("PatientUpdate.PNG"));
```

## Reports

I just changed the row height in the transaction logging report so that it stretches to fit the contents, which I know was a ticket at some stage but I can't find. Anyway, here's how to do this in future

- we should have auto-stretching rows for all reports, otherwise the user might not realise that the contents of some cells has been truncated.

1. Set all text fields and vertical lines to stretch type: relative to band height, and fix relative to top
2. Set all text fields to stretch with overflow (it's in the text field properties tab)
3. Set the bottom line of the band (underneath the text fields) to no stretch, but fix relative to bottom

| 14/01/2008, Melissa Loudon <melissa.loudon@gmail.com>

## General things to avoid

1. == only works for primitive types. You can't use it compare strings, even to check if a string is empty. So, taking this line of code from PharmacyInfo, which would fail to detect empty fields

```
#! java
if(txtCity.getText().trim() == "")
```

It should actually be:

```
#! java
if(txtCity.getText().trim().equals(""))
```

2. Misunderstanding the difference between SQL and HQL queries, which I've noticed a couple of examples of.

- Unlike SQL, you should probably never have a query with more than one entity type in the from clause
- if you do, ask yourself why there isn't a relationship between those entities.
- Also, prescription.patient refers to the actual patient, while prescription.patient.id refers to the patient's ID (the foreign key in the prescription table).

So, taking this query from StockManager.calculateStockLevels:

```
#! java
totalDrugsPackaged = ((Long) session.createQuery(
    "select Coalesce(sum(pd.amount), 0) "
    + "from Stock s, PackagedDrugs pd, Packages p "
    + "where s.id = :id " + "and pd.stock = s.id "
    + "and pd.parentPackage = p.id "
    + "and p.prescription is not null
").setInteger("id",
    theStock.getId()).uniqueResult()).intValue();
```

It should actually be:

```

#! java
totalDrugsPackaged = ((Long) session.createQuery(
        "select Coalesce(sum(pd.amount), 0) "
        + "from PackagedDrugs pd "
        + "where pd.stock.id = s.id "
        + "and pd.parentPackage.prescription is not null
").setInteger("id",
    theStock.getId()).uniqueResult()).intValue();

```

If you are unsure about how an SQL query translates to HQL, you should have a good look at the examples at [HQL query reference](#) and maybe also get someone else to check your code.

Also have a read over [Java antipatterns](#)

## dbscripts

### Database scripts

#### **idart\_x.x.0\_schema.sql**

Script to drop/generate the iDART database (with NO data)

1. Run the following pg\_dump command

```

pg_dump -h <host> -U <user> -F p -s -c -D -v -O -x -f
"idadt_x.x.0_schema.sql" idart

```

2. delete all the "ALTER TABLE ONLY <tablename> DROP CONSTRAINT <constraintname>;" statements

3. replace all the "DROP TABLE <tablename>;" statements with "DROP TABLE IF EXISTS <tablename> CASCADE;"

#### **idart\_x.x.0\_coredata.sql**

Script to populate tables with REQUIRED data only

1. Run the following pg\_dump command

```

pg_dump -h <host> -U <user> -f "idadt_x.x.0_coredata.sql" -d -a
--disable-triggers -t chemicalcompound -t chemicaldrugstrength -t clinic -t
drug -t form -t pharmacy -t regimen -t regimendrugs -t simpledomain -t
users idart

```

#### **idart\_x.x.0\_testdata.sql**

Script to populate tables with test data

1. run the following SQL commands on a database with real data:

```

update patient set firstnames = 'first'|| id;
update patient set lastname = 'last'|| id;
update patient set patientid = 'FN' || id;

```

2. use the following pg\_dump command

```
pg_dump -h <host> -U <user> -f "idart_x.x.0_testdata.sql" -d -a  
--disable-triggers -T chemicalcompound -T chemicaldrugstrength -T clinic -T  
drug -T form -T pharmacy -T regimen -T regimendrugs -T simpledomain -T  
users idart
```

## **idart\_x.x.x\_to-latest-sqldiff.sql**

Script to update a database from a base version to the latest version. It is made up of a set of self contained update procedures. For each change to the database a new section should be added to the end of the file.

### **Format**

This file is formatted as follows:

```
==== drop existing function ===  
  
==== comment section ===  
==== procedure definition to update to version x.x.1 ===  
==== call procedure to update to version x.x.1 ===  
. . .  
==== comment section ===  
==== procedure definition to update to version x.x.x ===  
==== call procedure to update to version x.x.x ===  
  
==== drop existing function ===
```

Each update section should look like this:

```

-----
-- iDART Datamodel version 3.0.x
-- Updater name (e.g. Simon Kelly)           update date (e.g. 04/07/2007)
-- Description of update
-----
DROP FUNCTION IF EXISTS diffprocedure(new_db_version VARCHAR);

CREATE OR REPLACE FUNCTION diffprocedure (new_db_version VARCHAR) RETURNS
boolean AS $$

DECLARE
doupdate boolean;
BEGIN
SELECT (REPLACE(value, '.', '0') < REPLACE(new_db_version, '.', '0')) into
doupdate FROM simpledomain WHERE name = 'database_version';
IF doupdate THEN
RAISE NOTICE 'Updateing to %', new_db_version;
-- START YOUR QUERY(IES) HERE --
-- END YOUR QUERY(IES) HERE --
UPDATE simpledomain SET value = new_db_version WHERE name =
'database_version';
END IF;
RETURN doupdate;
END;
$$ LANGUAGE plpgsql;

SELECT diffprocedure('3.0.x');

```

## hardware

### Standard Hardware

- Standard personal computer (minimum recommendation of 512Mb of RAM) running Windows or Ubuntu
- Zebra Label Printer (recommended TLP2844)
- Handheld barcode scanner (USB)
- Standard A4 printer (for reports)
- GSM modem with data-enabled SIM card (if an offsite backup is required). Cell-Life currently uses the [Maestro 100](#) GSM modem.
- Uninterruptable Power Supply (UPS)

### Consumables

- Labels – recommended: 75mm x 50mm standard vellum white labels
- Wax ribbon (out) – recommended 84mm width
- Paper and toner for standard A4 printer

### Quantities

As for quantities of labels and ribbons. The actual number of labels printed at a site depends on a number of factors such as the number of patients dispensed to, whether or not the site prints labels for non-ARV packages etc.

- The labels come in a roll of 1000.
- The ratio for determining ribbons has been estimated at 1 roll = 0.7 ribbons.

Some statistics from Cell-Life sites

Number of patients	Labels printed per month
3465	5300
2053	8000
3303	9000
2001	6100
1613	1700
2053	10000

## ideas\_concurrency

### Implementing Concurrency Control in iDART

#### Optimistic v. pessimistic locking

At the moment, all write transactions in iDART are enclosed in a transaction block, and the session is flushed after each transaction. The problem is that if two iDART instances are being used on one database, the concurrency strategy is automatically "last-commit-wins". It looks like this:

- To start, User A and User B both have a record (e.g. a patient) loaded
- User A makes some modifications to the record and saves them
- Then, User B makes some other modifications to the record and saves them
- The original record + User B's modifications are now saved. User A's modifications have been overwritten and lost.

To solve this, we can use either:

*Pessimistic locking:*

- the first user to load the record for modification locks the record, and no other user can load the record for modification until the lock is released by the first user

*Optimistic concurrency control:*

- Both users can load the record.
- User A modifies the record and presses save. iDART checks whether the copy of the record that User A was working on has been changed since it was loaded. It hasn't, so the save proceeds normally.
- User B modifies the record and presses save. iDART checks whether the copy of the record that User B was working on has been changed since it was loaded. It has, so the save is halted and a `StaleObjectStateException` is thrown. iDART asks the user whether they want to reload the changed object, or proceed and save over it. Or, iDART forces the user to reload the changed object.

For database updated in iDART, we should use Optimistic Concurrency Control. If chose pessimistic locking, we'd have to explicitly decide which objects will be locked, and when, and when the lock will be released

- I think this will become messy in code. For example, if you are the data quality manager updating a patient's details, should the pharmacist be prevented from dispensing to the patient while you are busy? Locking also becomes less appealing when you consider users who go off to do something else (make lunch or whatever) with a record locked.

Optimistic Concurrency Control requires much less code modification

- at a minimum, it's just adding a version number column to all our persistent classes (hibernate handles updating and checking the version number from then on) and catching and handling `StaleObjectStateExceptions` from our existing explicit transaction blocks. How the exception will be handled depends on what the operation is

- to start with, we could only offer the user the option to reload the modified object (losing any modifications they have made) or overwrite the modification with their changes.

Later, we could try merging the changes from the modified object with the changes the user has made on the screen for certain screens where the user may have done some work they'd be annoyed to lose. Bearing mind that iDART is generally low concurrency though, this is unlikely to happen often, and the work a user does before a transaction commits (so, the work they'd lose by reloading a modified object) is usually small. We'd want to work up to it, because modifying children causes the parent to be flagged as modified with a version number change (i.e. adding a prescription to a patient would cause the patient to be flagged as changed) but could definitely start off without it and not cause users much pain.

## Reading

As with most hibernate documentation, a lot of specific 3-tier web apps. Here are some useful links on optimistic concurrency control though:

[http://en.wikipedia.org/wiki/Optimistic\\_concurrency\\_control](http://en.wikipedia.org/wiki/Optimistic_concurrency_control)

[http://www.hibernate.org/hib\\_docs/reference/en/html/transactions.html#transactions-optimistic-longsession](http://www.hibernate.org/hib_docs/reference/en/html/transactions.html#transactions-optimistic-longsession)

and on pessimistic locking:

<http://www.postgresql.org/docs/8.1/static/sql-select.html#SQL-FOR-UPDATE-SHARE>

[http://www.hibernate.org/hib\\_docs/reference/en/html/transactions.html#transactions-locking](http://www.hibernate.org/hib_docs/reference/en/html/transactions.html#transactions-locking)

## Implementation Considerations

1. Ensuring concurrent transactions don't accidentally overwrite one another. This is where hibernate's in-built optimistic locking strategy works very well. If you add a version field to a persistent class, changes to the same persistent instance will throw a `StaleObjectStateException`, and will not overwrite each other. It is valid to exclude one or both sides in a one-to-many relationship from the optimistic locking strategy using the annotation `@!OptimisticLock(excluded=true)`. For an example, see Prescription. We want one user to be able to update a patient's attributes, while another user updates their prescription, so we use `@!OptimisticLock(excluded=true)` on the Patient property of Prescription.
2. Preventing the user from making decisions based on stale data. There are some transactions that do not update persistent objects, but require that a version check be conducted on them anyway prior to allowing the transaction to complete. An example is creating a package for a prescription. If the prescription has been updated to be inactive before the user saves the package, this is fine from hibernate's point of view - adding the package to the prescription does not overwrite the inactive property, and a version check is not needed. However, we don't want the user to be able to create a package for a prescription that has been changed. We can force a version check during the transaction that saves the package by calling `sess.lock(objectToVersionCheck, LockMode.READ)`. If the version check fails, a `StaleObjectStateException` will be thrown.
3. Preventing constraints that are not in the data model from being violated. An example of this is stock. We do not want to do a version check on the stock batch we are dispensing from every time we create a package, as this would be really annoying for 2 users dispensing from the same batch concurrently. So, we use `@!OptimisticLock(excluded=true)` on the Stock property of PackagedDrugs, and we do not force a version check when creating the package. The problem is that it is then possible for 2 users to both dispense the last unit of stock from the batch, leaving the batch at -1. I don't have a good solution to this within the transaction. One suggestion would be to check after the transaction has completed that the stock level is not negative, and force the user to delete and repackage, or perform a stock adjustment. This means that the transaction would not be atomic though. I do see a place for this solution if we are going to start looking at helping users maintain data quality in iDART - we could have negative stock batches as a data quality issue, and provide a way for users to see all problematic batches (along with other data quality problems). Generally I think it's unavoidable that concurrency will make us have to think harder about the constraints we enforce programmatically and not through the data model  
- we would have to be careful about taking these for granted as concurrent use might cause these constraints to be violated.
4. User Experience and Frustration. At first glance, optimistic locking seems like a less suitable strategy from the point of view of user experience, because users will occasionally make changes that, when they save, they will be told they have to redo.

- This will actually happen very rarely if we configure optimistic locking correctly
  - it should only force users to restart when two users are performing completely incompatible operations, like both updating the same prescription, or one updating a prescription that another is deleting. Facility process will generally make sure this doesn't happen.
- Operations in iDART on one screen generally take <1 minute to complete, so aren't too painful to redo.
- We could limit the annoyance of having to redo an operation by having a reload method that doesn't "lose" any of the information that hasn't changed.
- Pessimistic locking would result in far more situations where a user is prevented from doing their desired operation because another user has a lock on the data. For example, all a patient's prescriptions would have to be locked every time a user is updating the patient in case the patient update involves ending the episode (and consequently ending the prescription). But generally, these a patient update wouldn't involve changing anything about prescriptions, so the two operations would be able to happen concurrently.

## Suggested strategy

### Activity diagram with read and write sessions

All operations that update the database (whether creating new entities or updating existing entities) follow a similar pattern:

1. The screen opens

2. Any lists and combo boxes are populated
3. If updating, the user selects an entity or series of entities to update and the GUI is populated
4. User makes additions or changes
5. Save

I suggest that the read operations, right up until the save attempt, take place in a session that stays open during user think time. This session should be read-only (i.e. never flush to the database) so that there are no inadvertent DB writes where concurrent modification exceptions might not be correctly handled.

The read session would then be closed in the save method, and a write session opened immediately afterwards. The objects that need to be saved would be loaded into the write session, and the save would be attempted.

If the save is successful, the write session should be closed and the screen should either close or be cleared

If there is a concurrent modification exception, the user can choose to either continue, overwriting conflicting changes, or restart the screen *from the start of the operation*

- i.e, with the entities they selected earlier updated to reflect the modification, and the GUI populated with the updated values, but losing any changes they made. If you think about a patient being updated, this would mean that the correct patient would be loaded
- they wouldn't need to search or enter the patient ID again.

To do this, we would need to keep the entity (or entities

- in the merge patient screen, for example, there are two patients) selected by the user and used for populating the GUI. Then, when opening a new read session, we would need to load these into the session.

To demonstrate this in code, I've done the following:

Added version columns to all persistent classes.

Written wrapper classes for Session - ReadSession and WriteSession. The Read session never writes to the database because it is set to flush manually , and session.flush() is overridden so that it does nothing. So, our data access classes (whether we just use the current manager classes or go for a pattern of Data Access Objects), could specify which of these they take as a parameter (almost always a ReadSession), and we could completely avoid accidental DB writes.

Copied GenericGui and GenericFormGui into the classes TestGenericGui and TestGenericFormGui, and made some modifications so that they implement the activity diagram above. This mainly happens in the !cmdSaveWidgetSelected() method of TestGenericFormGui, which looks like this:

```

protected void cmdSaveWidgetSelected() {
    if (!fieldsOk())
        return;

    boolean success = false;
    if ((getReadSession() != null) && getReadSession().isOpen())
        getReadSession().close();

    setWriteSession(HibernateUtil.getWriteSession());
    Transaction tx = null;
    try {
        tx = getWriteSession().beginTransaction();
        attemptSave(getWriteSession());

        tx.commit();
        success = true;

    } catch (StaleObjectStateException se) {
        if (tx != null)
    }
}

```

```

tx.rollback();

getLog().warn("Concurrent modification during save.", se);
MessageBox m = new MessageBox(getShell(), SWT.YES | SWT.NO
| SWT.ICON_QUESTION);
m.setText("Concurrent Update");
m
.setMessage("The information you are updating has been changed by
another user
since you loaded the screen. Do
you want to continue, overwriting any new changes?");

switch (m.open()) {
case SWT.YES: {

    if ((getWriteSession() != null) && getWriteSession().isOpen())
    getWriteSession().close();

    try {
        setWriteSession(HibernateUtil.getWriteSession());
        tx = getWriteSession().beginTransaction();
        reloadData(getWriteSession());
        attemptSave(getWriteSession());
        tx.commit();
        success = true;
    } catch (HibernateException he) {
        success = false;
        if (tx != null)
            tx.rollback();
        getLog().error("Hibernate Exception in save operation", he);
        showSaveExceptionMessage();
    }
    break;

}
case SWT.NO: {

    setReadSession(HibernateUtil.getReadSession());
    try {
        tx = getReadSession().beginTransaction();
        reloadData(getReadSession());
        tx.commit();
        populateGUI();
        return;
    } catch (HibernateException he) {
        success = false;
        if (tx != null)
            tx.rollback();
        getLog().error("Hibernate Exception in data reload operation", he);
    }
}

}

```

```
}

catch (HibernateException he) {
    success = false;
    if (tx != null)
        tx.rollback();
    getLog().error("Hibernate Exception in save operation", he);
    showSaveExceptionMessage();

}
if (getWriteSession() != null && getWriteSession().isOpen())
    getWriteSession().close();

if (success) {
    MessageBox feedBack = new MessageBox(getShell(), SWT.OK
        | SWT.ICON_INFORMATION);
    feedBack.setText("Database Updated");
    feedBack.setMessage(getSuccessfulSaveMessage());
    feedBack.open();
    cmdCloseSelected();
} else
```

```
    cmdClearWidgetSelected();
}
```

The form itself then just needs to implement the abstract methods (some examples below), and not deal with session management at all:

```
@Override
protected void attemptSave(WriteSession wSession) {
    if (localClinic == null)
        localClinic = new Clinic();
    setLocalClinic();
    wSession.saveOrUpdate(localClinic);

}
```

```
@Override
protected void refreshData(Session sess) {
    localClinic = (Clinic) sess.load(localClinic.getClass(),
        localClinic.getId());
}
```

```
public void populateGUI() {

    txtClinic.setText(localClinic.getName());
    txtLocation.setText(localClinic.getAddress1());
    txtStreet.setText(localClinic.getAddress2());
    cmbProvince.setText(localClinic.getProvince());
    txtCity.setText(localClinic.getCity());
    txtPostal.setText(localClinic.getPostalCode());
    txtTelephone.setText(localClinic.getTelephone());
    txtClinicNotes.setText(localClinic.getNotes());

}
```

### What still needs to be considered

- Removal of LocalObjects lists (done in iDART > 3.1)
- Double-check relationships are modeled correctly, and decide (for each entity) whether there are any properties that should be excluded from the optimistic locking strategy
- Constraints enforced programmatically must not just be checked in the fieldsOk() method, as this could be working with stale data in a session. We should consider enforcing unique constraints (e.g. on patient id) in the data model, and deriving generated fields like prescription ID during save.
- Access to the database outside of Hibernate (e.g. by data import methods) needs to be controlled

- should these pessimistically lock?
- Separating data access and GUI creation code. This might only happen during the rewrite, but at least centralizing the session management in the generic GUIs should make it a bit easier.

## labelTesting

### Label Testing

There are a number of different labels in iDART, namely

- \* Package Cover Label
- \* Patient Info Label
- \* Drug Info Label
- \* Prescription Label ( or Summary Label )

There are many facets to each label which are able to be tested, and thus, here it is covered from each label, the areas targeted for testing:

#### Package Cover Label

- Tests:

- Checking if iDART and eKapa are similar in fields and looks.

There is no real test here, since data is properly obtained from Database.

#### Patient Info Label

- Tests:

- Check Patient Name Compression
- Date of Birth formatted to (dd MMM yyyy)
- Sex is written in full (Male, Female, Unknown)

#### Drug Info Label

- Tests:

- Add "Half" to Drug Label: Here when a quantity of 0.5 is added to a drug which is either a **tablet or a lozenge**, to denote that the patient should be taking just half a unit, iDART now should print the word "Half" instead of the number 0.5. Any other drug type will have the number 0.5, except as mentioned a tablet or a lozenge.
- Quantity dispensed + returned on drug label: iDART prints labels which denote the amounts that are being dispensed for each drug, including the amount of drugs that are left from the previous dispensed pack, which are named "Quantity On Hand", when the patient comes in to pick up their latest package, according to their appointment date which tells them when they should next pick up their drug.  
Unknown macro: {br}

> Here we test if the amounts of drugs print out as follows: a pack of 30 pills with 1 pill on hand should print out something like :

**D4T Stavudine 30mg (30 + 1)**

> Here we test if the amounts of drugs print out as follows: a pack of 30 pills should print out something like :

**D4T Stavudine 30mg (30)**

> Here we test if the amounts of drugs print out as follows: a pack of 90 pills with 4 months should print out something like :

**D4T Stavudine 30mg (90), D4T Stavudine 30mg (30)**

- each on its respective label

> Here we test if the amounts of drugs print out as follows: a pack of 90 pills with 4 months and 2 pills on hand should print out something like :

**D4T Stavudine 30mg (90 + 2), D4T Stavudine 30mg (30)**

- each on its respective label

- Combine eKapa and iDART labels: These are two different labels that are printed selectively, which are printed dependent on a flag in the properties file, are tested for their similarities, and if both have proper information being shown.
- Next Appointment on Drug Label: Only the Drug label and the Summary Label have the Next Appointment date showing, and this date can be configured to show or not to show.

#### Prescription Label

- Tests:

- Next Appointment: This label shows the summary of all the other labels that have been printed. There is a configurable field here, namely the Next Appointment Date, which is displayed according to a flag in the build.property file
  - Ensure this is configurable by changing the flag to true and false respectively and observing the results in the label, after printing.
- Quantity Dispensed, and Returned: Test if the proper printing of each drug name and its quantity dispensed and returned has been done. The above tests for quantity dispensed and returned refer to this test. Proper quantities reflect the above printed label quantities, hence this is a summary of the above printed labels. Ensure quantities of above printed labels corresponds to this summary/prescription label.

## **ltsp**

### **iDART on ThinClients with LTSP**

- iDART on ThinClients with LTSP
  - Additional steps
    - Set IP addresses for each client.
  - 2. Install iDART
  - 3. Configure the Label printers

1. Install and configure your LTSP Server ==  
See <https://help.ubuntu.com/community/UbuntuLTSP>

### Additional steps

#### **Set IP addresses for each client.**

In order to use the techniques below for printer configuration it is necessary to specify the IP addresses (and optionally the host names) for each client based on their MAC address. This can be done by adding to the `/etc/ltsp/dhcpd.conf` file as follows:

```
host workstation1 {
    hardware ethernet 00:00:00:00:00:00;
    fixed-address 192.168.0.200;
    option host-name "workstation1"
}
```

See [http://wiki.ltsp.org/twiki/bin/view/Ltsp/DHCP#Automatically\\_assigning\\_hostname](http://wiki.ltsp.org/twiki/bin/view/Ltsp/DHCP#Automatically_assigning_hostname) for more info.

## 2. Install iDART

Once you have your server running and you can boot into it over the network it is time to install iDART.

The recommended configuration is to install it to `/usr/local/iDART` and the create a shortcut for each user on their desktop. It is also suggested that you create an 'idart' group which all the users can belong to. To do this follow these steps:

1. Create the 'idart' user group.

Use the 'System ? Administration ? Users and Groups' dialog.

1. Run the iDART installer as root.

```
sudo java -jar idart-install-3.1.8.jar
```

When prompted for the install directory set it to `'/usr/local/iDART'`.

2. In order for users to be able to run the program you need to set the file permissions as follows:

```
sudo chown -R root:idart /usr/local/iDART
```

```
sudo chmod -R g+w /usr/local/iDART
```

3. In order to have a shortcut to iDART appear on the users desktop without having to manually copy one there for each new user you can place the shortcut in `/etc/skel/Desktop`

```
sudo mkdir /etc/skel/Desktop
```

```
sudo cp ~/Desktop/Cell-Life.desktop /etc/skel/Desktop
```

## 3. Configure the Label printers

Printing from an LTSP client is not as simple as printing from a normal desktop even if your printer is attached to the client machine. What you need to do is to configure your server so that it can see all the printers you have connected to all your clients. Having done that you can then tell the client which printer to use as its default printer.

iDART will try and use the default printer (as shown by the `lpstat -d` command). If it can not use that printer it will look for any Zebra printer. It is therefore important that it finds the right printer otherwise you will not be able to control which printer it prints to.

#### **1. Create a file in `/var/lib/tftpboot/ltsp/i386` called `ltsp.conf` with the following contents:**

```
Unknown macro: {ba} # Global defaults for all clients
ba # if you refer to the local server just use theba. # "server" keyword
as valuesba # see its parameters.txt for valid values.
ba #####ba. [default] PRINTER_0_TYPE=U
PRINTER_0_DEVICE=/dev/usb/lp0ba. }
```

#### **2. Add your Zebra printers.**

Use the 'System ? Administration ? Printing' dialog.

- a. Select 'New Printer'
- b. In the 'Select Connection' list choose '!AppSocket/HP JetDirect'.

- c. Type the IP address or hostname of the client to which the printer is attached and press the 'Forward' button.
- d. Leave the default option selected (Generic) and press the 'Forward' button.
- e. Leave the default option selected (text-only) and press the 'Forward' button.
- f. Give the printer a name i.e. ws1Zebra and a descriptions (Zebra printer on Workstation 1).
- g. Click the 'Apply' button.

Repeat these steps for all your Zebra printers.

### **3. Configure the clients to use the correct printer.**

#### **3.1. Unique user per client**

*The easiest way of doing this is to ensure that the same user logs onto each client every time. It is then possible to specify the default printer for that user.*

*In this case you would create users like 'ws1' for each workstation. Then for each of those users you would use the following command to specify their default printer:*

*'lpstat -v' (to list all the printers)*

*lpoptions -d printerName (you need to be logged in as that user)*

*For more info see: [LTSP default printer, and LTSP Printers](#)*

#### **3.2. Login script**

*The previous technique only works if the correct user logs onto the correct machine. A much better approach is to use a login script that checks to see which client the user is logged into and sets the default printer accordingly. To do this you need to modify the /etc/bash.bashrc file and add as section as follows (assuming you are using Bash and Ubuntu):*

```
case `echo $SSH_CLIENT | sed s/\.[[:digit:]]*\.[[:digit:]]*\$//` in
  192.168.0.200)  lpoptions -d Zebra1 > /dev/null ;;
  192.168.0.201)  lpoptions -d Zebra2 > /dev/null ;;
  192.168.0.202)  lpoptions -d Zebra3 > /dev/null ;;
esac
```

*Modify the cases to suit your IP addresses and printer names. The printer names can be seen by running the `lpstat -v` command.*

*Thanks to Jim McQuillan for this solution (<https://lists.ubuntu.com/archives/edubuntu-users/2007-January/000602.html>).*

## **network**

### **Installing iDART on a network**

In many situations it is convenient to have more than one iDART machine connected to the same database. This is easily achieved and involved four configuration steps in addition to the normal installation of iDART and PostgreSQL.

#### **1. Modify the postgresql.conf file to enable connections from other computers**

By default PostgreSQL will not accept connections from any computer other than the one it is installed on. To enable connections from other computers you need to edit the `postgresql.conf` file which can be found in the following locations:

Windows: C:\Program Files\PostgreSQL\8.3\data\postgresql.conf
Linux:

Change the following line as shown below (remove the '#' and change 'localhost' to '\*'):

```
#listen_addresses = 'localhost'
```

to

```
listen_address = '*'
```

## 2. Modify the pg\_hba.conf file

In order for other machines to be able to authenticate to the database server we need to add a line to the *pg\_hba.conf* file to specify the authentication method for the other machine. The *pg\_hba.conf* file can be found in the same location as the *postgresql.conf* file.

Edit the file and add the following lines to the end of the file and change the IP-ADDRESS and IP-MASK to match your network settings:

```
# TYPE DATABASE USER IP-ADDRESS IP-MASK METHOD
host all all 127.0.0.1 255.255.255.0 trust
```

## 3. Restart PostgreSQL

To apply the changes we made to the configuration files we need to restart the PostgreSQL server.

In Windows this is done as follows:

1. Logon to Windows with Administrator rights.
2. Click Start > Control Panel.
3. Double-click Administrative Tools.
4. Double-click the Services icon.
5. Double-click the service that you want (*postgresql*) to stop or start.
6. When the Service Properties window appears, do one of the following:
  - If the service is running, click Stop the service and then once it has stopped click Start the service.
  - If the service is not running, click Start the service.
7. Click OK.

## 4. Install iDART

In order to get iDART to connect to your remote database we must configure the database settings to tell it where the database is. The default value for the location of the database server in the iDART installer is *localhost*. This needs to be changed to be the IP address or hostname of the computer on which the database server is. Having done that install iDART as normal.

### Additional Resources

- PostgreSQL pg\_hba.conf reference
- PostgreSQL manual

### svn\_merges

### SVN Merge tracking

- SVN Merge tracking
  - Into trunk
    - 20/08/2009 from iDART-3.4
    - 12/06/2009 from iDART-3.4
    - 17/04/2009 from iDART-3.4
    - 06/04/2009 from cd4
    - 25/02/2009 from 3.3
    - 03/07/2008 from stockCenter
    - 23/06/2008 from 3.2
    - 26/05/2008 from 3.1
    - 14/05/2008 from 3.1
    - 14/05/2008 from 3.1
    - 14/05/2008 from 3.1
    - 25/04/2008 from 3.1
  - Into iDART-3.2 branch
    - 20/06/2008 from 3.1
    - 20/06/2008 from 3.1
    - 10/06/2008 from 3.1
    - 10/06/2008 from 3.1
  - Into stockcenter branch
    - 23/06/2008 from 3.2
  - Into iDART-3.1 branch
    - 18/04/2008 from trunk

This page is to help track merges that happen between SVN branches.

## Into trunk

### **20/08/2009 from iDART-3.4**

- Reason: merge from branch
- Merge by: Simon
- Merge: [branches-iDART-3.4@4447.4457](#)
- Resulting changesets: [4459](#)

### **12/06/2009 from iDART-3.4**

- Reason: merge from branch
- Merge by: Simon
- Merge: [branches-iDART-3.4@4419.4447](#)
- Resulting changesets: [4448](#)

### **17/04/2009 from iDART-3.4**

- Reason: merge from branch
- Merge by: Simon
- Merge: [branches-iDART-3.4@4413.4418](#)
- Resulting changesets: [4419](#)

### **06/04/2009 from cd4**

- Reason: merge from branch
- Merge by: Simon
- Merge: [cd4 branch](#)
- Resulting changesets: [4409](#)

### **25/02/2009 from 3.3**

- Reason: merge from branch
- Merge by: Simon
- Merge: [iDART-3.3@4216:4384](#)
- Resulting changesets: [4387 4388 4389](#)

### **03/07/2008 from stockCenter**

- Reason: new feature completed
- Merge by: Rashid
- Merge: [iDART-stockCenter@3776:3857](#)
- Resulting changesets: [3860](#)

### **23/06/2008 from 3.2**

- Reason: bugfixes
- Merge by: Simon
- Merge: [iDART-3.2@3525:3766](#)
- Resulting changesets: [3769 3770 3771](#)
- excluded revisions r3764, r3765

### **26/05/2008 from 3.1**

- Reason: bugfixes
- Merge by: Rashid
- Merge: [iDART-3.1@3450:3522](#)
- Resulting changeset: [3524](#)

**14/05/2008 from 3.1**

- Reason: bugfixes
- Merge by: simon
- Merge: [iDART-3.1@3417:3449](#)
- Resulting changeset: 3452
- excluded revision 3444 from iDART-3.1 branch
- excluded revision 3416 from iDART-3.1 branch

**14/05/2008 from 3.1**

- Reason: bugfixes
- Merge by: simon
- Merge: [iDART-3.1@3352:3401](#)
- Resulting changeset: 3450
- excluded revision 3351 from iDART-3.1 branch : this problem was related to local objects and since local objects were removed from 3.2, this bug is not valid for 3.2

**14/05/2008 from 3.1**

- Reason: bugfixes
- Merge by: simon
- Merge: [iDART-3.1@3349:3350](#)
- Resulting changeset: 3448

**25/04/2008 from 3.1**

- Reason: bugfixes
- Merge by: simon
- Merge: [iDART-3.1@3277:3327](#)
- Resulting changeset: 3348

**Into iDART-3.2 branch**

**20/06/2008 from 3.1**

- Reason: bug fixes
- Merge by: Rashid
- Merge: [iDART-3.1@3750:3758](#)
- Resulting changeset: 3761

**20/06/2008 from 3.1**

- Reason: bug fixes
- Merge by: Rashid
- Merge: [iDART-3.1@3656:3749](#)
- Resulting changeset: 3761

**10/06/2008 from 3.1**

- Reason: bugfixes
- Merge by: simon
- Merge: [iDART-3.1@3626:3655](#)
- Resulting changeset: 3661

**10/06/2008 from 3.1**

- Reason: bugfixes
- Merge by: simon
- Merge: [iDART-3.1@3449:3625](#)
- Resulting changeset: 3660

## Into stockcenter branch

**23/06/2008 from 3.2**

- Reason: refactoring
- Merge by: simon
- Merge: [iDART-3.2@3763:3765](#)
- Resulting changeset: [3775](#)

## Into iDART-3.1 branch

**18/04/2008 from trunk**

- Reason: bugfixes
- Merge by: simon
- Merge: [trunk@3271:3274](#)
- Resulting changeset: [3277](#)

## user\_docs

### Editing the iDART User Manual and JavaHelp

The iDART user documentation is available both as a pdf book, and as JavaHelp that can be accessed directly from iDART.

For the rest of this howto, your working directory is the doc/ subdirectory of the iDART project in the workspace.

The source for both types of documentation is the iDART\_manual\_book.xml file. This is a docbook xml file To edit the text or structure of the iDART user documentation, you should edit this file. You shouldn't need to use many more syntax elements than are already used in the manual, but the complete docbook guide is [here](#) if you're feeling brave.

You will need to install some docbook packages:

```
sudo apt-get install docbook docbook-utils docbook-xml docbook-dsssl  
docbook-xsl
```

The images in the iDART user documentation are screenshots taken in Ubuntu with the default human theme (the brown colour scheme). They have been resized using [imagemagick](#) to be maximum 600x400 pixels. The script doc/images/resize.sh will resize all the images in that directory. You will need to have imagemagick installed

- on ubuntu,

```
sudo apt-get install imagemagick
```

Once you have the images resized correctly, you need to convert the iDART\_manual\_book.xml file into the pdf userguide, and the generated javahelp. You need [apache fop](#) (version 0.25) installed to do this.

To generate the pdf use manual, you first need to set the correct path to the fo/docbook.xsl stylesheet in iDARTfo.xsl. Edit this file in a text editor. Then:

```
$ cd workspace/iDARTproject/doc/  
$ xsltproc iDARTfo.xsl iDART_manual_book.xml >iDART_manual_book.fo
```

Set JAVA\_HOME to point to a JRE. This JRE needs to have Java Advanced Imaging (JAI) installed. You can download JAI [here](#).

```
$ JAVA_HOME=/home/melissa/programs/udig/jre/;export JAVA_HOME
```

Put the directory containing fop.sh on the PATH:

```
$ PATH=$PATH:/home/melissa/DocumentationTools/fop-0.20.5; export PATH
```

Then, generate the pdf manual

```
$ fop.sh -fo iDART_manual_book.fo -pdf iDART_manual_book.pdf
```

To begin generating the javahelp, Set the correct path to the javahelp/javahelp.xsl stylesheet in iDARThelp.xsl. Edit this file in a text editor. Then, generate the javahelp with:

```
$ cd workspace/iDARTproject/doc/
$ xsltproc iDARThelp.xsl iDART_manual_book.xml
```

This produces normal chunked html output as well as the javahelp helpset metadata, so you should be able to view the manual as multipart html by opening index.html in a web browser

All that remains is to index the helpset so that it can be searched. To do this, you will need the JavaHelp system, which you can download from <http://java.sun.com/products/javahelp/>.

To index the helpset:

- Download and unzip the JavaHelp system
- Delete any files in doc/JavaHelpSearch (but not the whole directory, or the hidden .svn directory, as this will confuse eclipse when you try to commit)
- create the index files by issuing the following command, while in the doc/ directory

```
/path/to/jh2.0/javahelp/bin/jhindexer .
```

## Release process

To create a new release of iDART follow the following procedure:

### Maintenance release

A maintenance release is a new release from an existing branch that fixes some bugs but shouldn't have any new features in it.

1. Checkout the branch from SVN
2. Run the *DORERELEASEAnt* build task
  - a. This should fail at the end and ask you to update the readme.md file with the changelog and upgrade notes
3. Copy the changelog and upgrade notes from the previous release and add a new section for the current version.
4. Run the *ziprelease* Ant build task
5. Copy the release zip to \\biko\\cell-life.shared.idart\\Releases
6. Upload the release to [SourceForge](#)
  - a. Don't forget to mark the new release as the default download.
  - b. See [SourceForge help](#) for more info.
7. Notify Oliver of the release so that he can send out a notification via the website about the new release.
8. Increment the *idart.version.revision* property in the *build.properties* file and commit it to SVN.

## Major release

1. Create a new release branch with a name of iDART\_X.Y where X and Y are the major and minor version numbers of the release.
2. In the *trunk* increment the *idart.version.minor* property in the *build.properties* file to the next version and commit it.
3. Follow the maintenance release process to release the version from the new branch.

## Screens

### Clinic

iDART: Add New Clinic (logged in as admin)

### Add New Clinic

All fields marked with \* are compulsory

\* Clinic:

Telephone:

Notes:

**Location**

Country:

Province:

District:

Sub-District:

Facility Name:

Facility Type:



iDART: Update an Existing Clinic (logged in as admin)

## Update an Existing Clinic

All fields marked with \* are compulsory

* 'Clinic:	1349995397779	Clinic Search
Telephone:	27215556666	
Notes:	Things and Stuffs	

**Location**

Country:	South Africa	
Province:	Western Cape	Search National List
District:	West Coast DM	
Sub-District:	Swartland LM	
Facility Name:	Oaklahoma ARV Site	
Facility Type:	Clinic	

**Buttons**

Save    Clear    Cancel

### Inputs

- Text
  - ClinicName
  - ClinicTelephone
  - ClinicNotes
  - FacilityType
- Combo
  - Country
  - Province
  - District
  - SubDistrict
  - FacilityName
- Button
  - NationalClinicSearch

### Delete Stock/Prescriptions/Packages

iDART: Stock, Prescription and Package Deletions (logged in as admin)

## Stock, Prescription and Package Deletions

Undo Created Package     Redo Single Item in Package     Delete Incorrect Stock Batch     Delete Incorrect Prescription

Patient Nu:  Patient Search

Package To Remove:

Package ID:   
Date Packed:

Drugs in This Package:

Dispensed / Accum	Drug Name	Qty	Batch No

iDART: Stock, Prescription and Package Deletions (logged in as admin)

## Stock, Prescription and Package Deletions

Undo Created Package     Redo Single Item in Package     Delete Incorrect Stock Batch     Delete Incorrect Prescription

Patient Nu:  Patient Search

Package To Remove:

Package ID:   
Date Packed:

Drugs in This Package:

Dispensed / Accum	Drug Name	Qty	Batch No



## Stock, Prescription and Package Deletions

 Undo Created Package Redo Single Item in Package Delete Incorrect Stock Batch Delete Incorrect Prescription

Drug:

Drug Search

Available Batches for Drug:

Date Received	Qty	Batch No	Manufacturer	Expiry Date	Pharmacy
---------------	-----	----------	--------------	-------------	----------

Remove Selected Batch

Clear

Close



## Stock, Prescription and Package Deletions

 Undo Created Package Redo Single Item in Package Delete Incorrect Stock Batch Delete Incorrect Prescription

Patient Nru:

Patient Search

Prescription To Remove:

Prescription ID:

Date Captured

Drugs on the Prescription:

Drug Name	Dosage
-----------	--------

Remove this Prescription

Clear

Close

## Destory Stock

iDART: Destroy Unusable Stock (logged in as admin)

 Destory Unusable Stock

Select a Pharmacy: Main

Drug Information

Drug Name:  Drug Search

Total Packs in Stock:

One Pack Contains:

Batch Information

The amount received, dispensed, destroyed and in stock for each batch is shown in packs with the number of pills in brackets  
'10 (5)' means 10 packs and 5 loose pills, while '10' means 10 packs and no loose pills.

Note: this table does not show stock adjustments but if a stock take was done then the 'In Stock' value will take this into consideration.

Batch	Shelf	Manufacturer	Expiry Date	Received	Dispensed	Destroyed	In Stock	Packs to Destroy	Loose Pills to Destroy

\* Reason for Disposal

Save Clear Cancel

## Patient Packaging

iDART: Make Up Drug Packages for Patients (logged in as admin)

## Make up Drug Packages for Patients

Patient Search:

**1. Search Results**

- 123456789 (Soap, Joe)
- 123457 (Skywalker, Luke)

**Patient's Clinic**

- Main Clinic

I am dispensing directly to patients

Dispensing from pharmacy: Main

I am creating packages for later pickup

Patient Information

Patient Number:

Patient's Name:

Patient's Age:  (03 Jun 2004)

Next Appointment:

This is package number  
1 of 1 month prescription

Package Contains:  supply

Date Packed:

Prescription Information

Prescription ID:

Doctor:

Date of Last Pickup:

Prescription Date:

Prescription Notes

Print next appointment date on label  Yes  No

Print a script summary label  Yes  No

Undo Created Package
 Create Package

### Inputs

- Text
  - SearchBar
  - PrescriptionID
  - PrescriptionNotes
  - PrescriptionDate
  - Doctor
  - NextAppDate
  - PatientID
  - PatientName
  - PatientAge
  - PatientDOB
- Combo
  - Supply
  - StockCenter

### Events

- Button
  - CaptureDate
    - Selection
  - Close
    - Selection
  - Dispense
    - Selection
  - DispenseNow
    - Selection
  - DispenseLater
    - Selection
  - PatientHistoryReport

- MouseUp
- RedoPackage
  - Selection
- ReprintLabels
  - Selection
- PrintCustomLabel
  - Selection
- UpdatePrescription
  - MouseUp
- Combo
  - SelectStockCenter
    - Selection
  - StockInHand
    - Modify
    - Focus
  - Supply
    - Selection
- Link
  - StockOnHand
    - Selection
- List
  - WaitingPatients
    - Selection
- Table
  - PrescriptionInfo
    - Selection
  - PillCount
    - Change
- Text
  - DrugLabel
    - Modify
  - SearchBar
    - Key
- Shell
  - DeleteStockPrescriptionsPackages
    - Dispose
  - BatchInformation
    - Dispose
  - AddPrescription
    - Dispose

## Stock Center

iDART: Add a New Pharmacy (logged in as admin)

## Add a New Pharmacy

Pharmacy Details (related to where stock is held)

Add New Pharmacy     Update a Pharmacy     Update Facility Detail

Pharmacy name

Preferred Pharmacy?  Yes  No

Facility Details (shown on labels and reports)

All fields marked with \* are compulsory

\* Facility Name:

\* Street Address:

\* City:

\* Telephone Number:

\* Head Pharmacist:

Pharmacy Assistant:

Preview of Label

Facility Name
Pharmacist
Physical Address

**Save** **Clear** **Cancel**

Update a Pharmacy

 Update a Pharmacy

Pharmacy Details (related to where stock is held)

Add New Pharmacy  Update a Pharmacy  Update Facility Detail

Pharmacy name  Pharmacy Search

Preferred Pharmacy?  Yes  No

Facility Details (shown on labels and reports)

All fields marked with \* are compulsory

\* Facility Name:

\* Street Address:

\* City:

\* Telephone Number:

\* Head Pharmacist:

Pharmacy Assistant:

Preview of Label

Facility Name
Pharmacist
Physical Address

Save  Clear  Cancel

Update Facility Details



## Update Facility Details

**Pharmacy Details (related to where stock is held)**

Add New Pharmacy     Update a Pharmacy     Update Facility Detail

Pharmacy name:

Preferred Pharmacy?  Yes  No

**Facility Details (shown on labels and reports)**

All fields marked with \* are compulsory

* Facility Name: <input type="text"/>	Facility Name
* Street Address: <input type="text"/>	Street Address
* City: <input type="text"/>	City
* Telephone Number: <input type="text"/>	Tel
* Head Pharmacist: <input type="text"/>	Demo Pharmacist, B.Pharm
Pharmacy Assistant: <input type="text"/>	Demo Pharmacist2, B.Pharm

**Preview of Label**

Facility Name  
Demo Pharmacist, B.Pharm  
Street Address, City, Tel

## SPRINT Meeting Notes

- 18 May 2010
- 01 June 2010
- 29 June 2010
- 13 July 2010
- 09 Sept 2010

### 18 May 2010

#### This SPRINT (Next 2 Weeks):

- Bug fixes
- BAs to meet with Siki to define Siki's reporting requirements (Send routers, not modems - always online, can be configured to connect to Cell-Life servers only - no general internet access, requires some development - rather send them the statistics from the PEPFAR report. Client will also want to get reports though)
- Simon create new milestone on Jira - 3.5.1
- Look at old Trac tickets (Rashid) & old tickets on Jira (Zahir)
- Document comparing features - Rashid to update the Wiki. Current - all drugs, nappi coding (link to external database), multiple prescriptions, SMS functionality
- Have a CIDA Tech Workshop

#### Notes:

- Rewrite iDART: Biggest issue is episodes and packaging
- Product Backlog is currently on Trac and we'll leave it here for the time being.

### 01 June 2010

#### This SPRINT (01 June - 15 June 2010)

- Setting up a VPN - researching how one does that
- Need a router - SK and SB to check budget
- Set up a VPN on the server
- Spec out the communication between the two (possibly web services, research the options)
- Continue with bug fixing if they arise
- BAs to meet with Siki to define Siki's reporting requirements (Send routers, not modems - always online, can be configured to connect to Cell-Life servers only - no general internet access, requires some development - rather send them the statistics from the PEPFAR report. Client will also want to get reports though)
- BAs to create GUI mockups for Data Quality Issue (output likely a CSV file). SK to do UML diagrams. Adiel to work with this.
  - Can select a test. Might have parameters. Quite like an export. Run script and dump it.
- If time available, Rashid to look at the episode ticket - SB to write out the ticket in more detail.

#### **General Notes**

- Need to create interface between the two systems
- Need idea of what functionality is needed
- e.g. Adherence reminders: Sent per patient, group of patients, schedules, contact details?
- **Basic functionality for communications between the two platforms**
  - Will have to send authorisation details - Payment for message
  - Adherence Messages: Number, time to be sent, message content, frequency (twice a day), start and end dates
  - Confirmation from Mobilisr - Query message statuses (iDART would do the query, Mobilisr wouldn't send data) - message ID
  - Appointment Reminders: Number, time to send, message content, once-off (only once an appointment has been made) + confirmation of receipt
  - Drug Changes: number, time to send, message content, frequency, start and end dates
  - PMTCT:
    - Scheduling single message and scheduling a recurring message (same content each time)
    - Send to individual or multiple
- **iDART:**
  - Be able to get a group (e.g. all men between ages of X and Y, all pregnant women)
  - Able to send message to a group (even all)
- Where should the content be put? e.g. PMTCT
- Need to know from a messaging point of view - interface between the two

#### **iDART Rewrite - Need Priorities:**

- 1 Dispensing / Packaging
- Label Printing

#### **Notes from Research Meeting with Katherine:**

- Have to include the opt-out (ethical)
- From research - want to find out why someone opts out - person would be called
- Opt-in - ethical considerations (Work flow in the pharmacy) - would need signed ethical consent + info sheets
- From iDART, they would just tick what streams they want

## **29 June 2010**

#### **This SPRINT (29 June - 13 July 2010)**

Due to Sarah being ill and Simon leaving for Europe, we didn't have an official SPRINT Planning session two weeks ago for this current sprint. Simon and Rashid did discuss it, and agreed to include the following work in this SPRINT:

- Rashid to finish off VPN - done (on dev server)
- Rashid to meet with Vikram re comms between the two (decided to go with web services, Vikram provided some APIs, Rashid researched which tools to use. Mobilisr doesn't have that functionality yet.) - done
- Adiel - bug fixes and Data Quality screen.
  - 80% complete, need discussions (SB to update ticket with consistent test names)
  - Rashid - concern over complexity required since simple script can be dumped - to speak with Simon

#### **THIS SPRINT:**

- Zahir - create one ticket for all outstanding test case tickets
- Sarah - buy router

- Rashid & Adiel - start development on web services
- Rashid - doc for Simon re APIs - email to get confirmation
- Rashid - changelog for 3.5.2 (but very likely will give RHRU GP 3.5.1)
- Sarah - go through critical Bug fixing on Jira
- Adiel - rewrite Data Quality methods to do basic scripting, not to go through the export (likely one day)
- Sarah - test data quality screen
- BAs to start thinking about screens for SMS functionality - how do we add someone to a program for example? Difficult as it relies a little on researchers.
- From previous SPRINT: If time available, Rashid to look at the episode ticket - SB to write out the ticket in more detail.

#### **General Notes**

- Need a stable release for RHRU GP - Get feedback from Raiyaan regarding the 1st label not printing, Also check Masi can print to PDF
- What is going to be sent to Mobilisr (Message, Content, Frequency of message)
  - Will have to send authorisation details - Payment for message
  - Adherence Messages: Number, time to be sent, message content, frequency (twice a day), start and end dates
  - Confirmation from Mobilisr - Query message statuses (iDART would do the query, Mobilisr wouldn't send data) - message ID
  - Appointment Reminders: Number, time to send, message content, once-off (only once an appointment has been made) + confirmation of receipt
  - Drug Changes: number, time to send, message content, frequency, start and end dates
  - Scheduling single message and scheduling a recurring message (same content each time)
  - Send to individual or multiple
- **iDART:**
  - Be able to get a group (e.g. all men between ages of X and Y, all pregnant women)
  - Able to send message to a group (even all)
- Where should the content be put? e.g. PMTCT
- Need to know from a messaging point of view - interface between the two

#### **iDART Rewrite - Need Priorities:**

- 1 Dispensing / Packaging
- Label Printing

#### **Notes from Research Meeting with Katherine:**

- Have to include the opt-out (ethical)
- From research - want to find out why someone opts out - person would be called
- Opt-in - ethical considerations (Work flow in the pharmacy) - would need signed ethical consent + info sheets
- From iDART, they would just tick what streams they want

## **13 July 2010**

#### **SPRINT Notes**

##### **This SPRINT (13 July - 26 July 2010):**

- Bug Fixes:
  - From previous SPRINT: If time available, Rashid to look at the episode ticket - SB to write out the ticket in more detail.
  - Zahir: Log ticket regarding default value for 'Dispense Directly' or 'Create package for later pickup'
  - Bug re syrup (2 separate tickets) - speak with Raiyaan, Raiyaan to test
  - Sarah - go through critical Bug fixing on Jira
  - Sarah - test data quality screen
- Release 3.5.2 at end of this SPRINT (includes data quality)
- Define the web service spec & start implementing it (on mobilisr first, then on iDART)
- BA to think through screens for enrolling 'new patient' on ARVs
- BAs to do testing (Complete and JIRA tickets)

#### **General Notes**

- What is going to be sent to Mobilisr (Message, Content, Frequency of message)
  - Will have to send authorisation details - Payment for message
  - Adherence Messages: Number, time to be sent, message content, frequency (twice a day), start and end dates
  - Confirmation from Mobilisr - Query message statuses (iDART would do the query, Mobilisr wouldn't send data) - message ID
  - Appointment Reminders: Number, time to send, message content, once-off (only once an appointment has been made) +

- confirmation of receipt
- Drug Changes: number, time to send, message content, frequency, start and end dates
- Scheduling single message and scheduling a recurring message (same content each time)
- Send to individual or multiple
- **iDART:**
  - Be able to get a group (e.g. all men between ages of X and Y, all pregnant women)
  - Able to send message to a group (even all)
- Where should the content be put? e.g. PMTCT
- Need to know from a messaging point of view - interface between the two

#### **iDART Rewrite - Need Priorities:**

- 1 Dispensing / Packaging
- Label Printing

#### **Notes from Research Meeting with Katherine:**

- Have to include the opt-out (ethical)
- From research - want to find out why someone opts out - person would be called
- Opt-in - ethical considerations (Work flow in the pharmacy) - would need signed ethical consent + info sheets
- From iDART, they would just tick what streams they want

## **09 Sept 2010**

### **General**

- iDART lacking direction
- Three 'chunks' of work that we can work on, but don't want to interchange regularly between these.
- Decided to focus on rewrite initially

### **iDART 3.5.2**

This is ongoing:

- Not printing 1 drug label (currently getting support to phone all the sites)
- Repeats, issues bug: Request from CAPRISA to fix this, Siki has approved. Zahir to write out business rules to fix this for the common case only (two months supply)
- Sarah to test the clinics, districts link up

### **iDART CIDA**

This will be put on hold (for possibly two months, or less), pending:

- CIDA Researcher starting
- BRS compiled (from interviews with key stakeholders)
- A functional Mobilisr

### **iDART Rewrite**

Decided to concentrate on this now, for at least 1 month, preferably two.

- Build
- Demo
- Label printing framework / API (need input from BAs to do this - a generic template - that the users can say what info they want in the different areas. The system will then adapt it to suit the label size).
- Packaging screen needs to be spec'd out (dispensing now or later, pill counts, issues and repeats, issue 2 weeks supply)
- Reporting framework / API
- Spring integration
- GUI framework - refactor
- Unit testing

### **Other Work (TB Integration)**

- Zahir currently working on the BRS (GF Jooste contacted, need to now contact Desmond Tutu TB Centre to get additional requirements)

- Decided to reevaluate this after the BRS is complete (possibly one month or less). Decide then which version it should be included in.

## New Features (3.6.0)

These features have been identified by Siki.

- Transfer bulk stock to other clinics (including invoices / receipts / reports)
- Costings - more comprehensive costing reports

## Wish List

1. Dispense database functionality needs to be mapped for changes to iDART

2. iDART the- 'ADD A NEW PATIENT SCREEN'. The option to take a picture with webcam. Can appear as a small pic but when cursor on it then automatically enlarges. In the bottom of the screen the option to "Print Patient Info ID Card" next to "Print Patient info Label" \*\*\*

Also following from this " Update a prescription Screen" and leading to "Add a Drug to Prescription" can we also include the same action as above requesting whether it should be a repeat or a single prescription. \*

5.DART- "Stock Arrives At Pharmacy" screen The end column has Unit Price. It would be great to add Total Price calculated by Unit price x Units received hence giving a total value of drugs received. \*\*\*

In the "Drug Dispensed Report" can we add the value amount \*\*\*

In the " Monthly Receipts and Issues Drugs Report" After the Stock on Hand Column can we add on the Price Value of the Drug \*\*\*

6.iDART- " Create Drug Group Screen" to "add a New Drug Group"- under Drug Group Information can we modify \* Regimen to allow us to add a number of other regimens? and not be restricted. \*

7.iDART- Patient visits Report under screen "Patient visits and Stats Module" if report can be seen on screen instead of only been saved.

8.iDART- under " patient visits and Stats Module" can we have the option " Patient Stats" to add on more under statistics as well as clinical records \*\*\*

10..iDART- modification of skin to add the new logo \*

11. additional screens for scripting acute and chronic meds.

12. Addition of ICD-10 directory for diagnosis under Medical records module.

11.if the iDART csv export utility could be enhanced to "quote" text strings, then Excel would import the Id number column correctly when you double-clicked.\*\*

12.Addition of nappi codes and option to add other product codes in the Drug module when we add new drugs.

13. Web based iDART should have the ability to backup database locally as well as central database.

14. Allow importing of (country, province districts. subdistricts and facilities in the different countries). Go to Gen Admin, then go " Add a New Clinic" and under location the ability to import the list according to each country

15. Create a new " Medical Records Module" and then bring "Patient visits and stats' under the new "Medical Records Module"

16. Under "Reports" then to "Missed Appointment Reports" Add another column "consented to be contacted:Yes/No" in the column next to the phone number. Synchronised to send only appointment reminders to those consented Yes

17. Open to "Gen Admin" then Open to "add a New Drug" the under drug is: o Side Treatment o ARV Drug - .....then Add the new following o TB Drug

18. Under Reports: Go to " Monthly issues and receipts" a separate page for TB drugs and Side Treatment

19. Under Reports: Drug Usage reports: one report on ARV's, one for TB treatment, one on Other drugs

20. Under Reports under" Clinic Indicator report" add the Number of patients on TB treatment

21. Based on the outgoing stock every month the system should calculate the average monthly consumption and report on this, buffer stock in the number of months, lead time in months, period to cover each order. The system shpuld flag in the daily stock overview .....stock $\leq$ AMC x (buffer stock+lead time). When flagged to send SMS message to pharmacist or persons responsible

22. Create order sheet to place order based on **AMC x ( buffer + period to cover with order + lead time) - stock**

23. Translate iDART into Portuguese, Spanish, French