DELIVERABLE 3.4

Database infrastructure for enabling studies based on prospective data in WP1 and 2

Grant agreement no.: 601055 (FP7-ICT-2011-9)
Project acronym: VPH-DARE@IT
Project title: Dementia Research Enabled by IT
Funding Scheme: Collaborative Project
Project co-ordinator: Prof. Alejandro Frangi, University of Sheffield
Tel.: +44 114 22 20153
Fax: +44 114 22 27890
E-mail: a.frangi@sheffield.ac.uk
Project web site address: http://www.vph-dare.eu

Due date of deliverable	Month 24
Actual submission date	Month 26
Start date of project	April 1st 2013
Project duration	48 months

Work Package & Task	WP 3, Task 3.2
Lead beneficiary	STH
Editor	PMO
Author(s)	Kevin Teh, Steven Wood
Quality reviewer	Annalena Venneri, Sebastien Kozerke

Project co-funded by the European Union within the Seventh Framework Programme		
Dissemination level		
PU	Public	**X**
PP	Restricted to other programme participants (including Commission Services)	
RE	Restricted to a group specific by the consortium (including Commission Services)	
CO	Confidential, only for members of the consortium (including Commission Services)	

Issue Record

Version no.	Date	Author(s)	Reason for modification	Status
1.0	01/03/15	K Teh	Initial release	Draft
1.1	26/04/15	S Wood	Final version following review comments	Complete
1.2	8/5/515	PMO	Final Check	Finalised

Copyright Notice

Author(s) for Correspondence

Lead author name
Correspondence details

Dr Steven Wood
T: +44 114 271 3649; E: steven.wood@sth.nhs.uk;

Table of Contents

Table of Figures

INTRODUCTION

VPH-DARE@IT is a hugely ambitious scientific endeavour seeking to integrate information from across many domains to provide:

> *"a systematic, multifactorial and multi-scale modelling approach to understanding dementia onset and progression and enable more objective, earlier, predictive and individualised diagnoses and prognoses of dementias to cope with the challenge of an ageing European society."*

The number of participants involved in the project and their disparate needs for scientific analysis results in a very complex collection of requirements. The two key components in an infrastructure to support such an activity are data storage and compute services and these two facets of the platform need to be integrated seamlessly to deliver a coherent solution.

A further separation in the data provision for the project is that of retrospective data collection available both publically and through consortium partners and prospective data collected during a targeted clinical trial executed within the project. Deliverable 3.1 deals with the former subtype, and this deliverable will describe in detail the infrastructure components put in place to support these data collection activities and also their integration with the rest of the data network.

1. BACKGROUND

The proposal outlined two possible solutions for clinical data capture and management available to the project, the first being a product produced and sold by Sheffield Teaching Hospital NHS Foundation Trust called ArQ. This is a complete clinical application development system which can be used to produce a highly integrated patient management system within the clinical environment. It is used extensively in Sheffield and is now also a component in several other clinical trials type scenarios several of which are EU funded projects.

There is no question that this system could be used to collect and manage the data associated with this project, however it is not open source, and it is not freely available. Whilst STH would be happy to allow its use free of charge for the duration of the project, there would have to be a funding stream to pick up the commercial aspects of continuing this system following the end of the project. In addition the real advantages of the ArQ system, i.e. its large collection of components for integrating data within the clinical environment, were not really necessary when we analysed the processes that the research teams would put in place to collect the data. Finally given the project philosophy of wanting to create a lasting and open platform, which may well desire additional data capture after the funded period, we decided that pursuing a proprietary solution which would create a kind of "vendor lock-in" was not desirable.

Following these considerations it was decided that ArQ would only be considered as the data management platform for the project once all other viable options had been ruled out.

From an open source and community supported software perspective this leaves us with two primary systems for consideration, OpenClinica (https://www.openclinica.com/) and XNAT (http://www.xnat.org/). These choices come from the almost equal demand in the project for an imaging based data collection, which is the primary function of the XNAT system, and the need to collect detailed clinical and phenotypical information

of the subjects for which OpenClinica is the largest community based offering. We decided to evaluate both these systems for use in VPH-DARE@IT and the following describes the implementation of the final system design.

2. SYSTEM DESIGN AND RATIONALE

2.1. OPENCLINICA EVALUATION

OpenClinica has excellent facilities for creating and managing electronic case report forms (eCRF's), and it enables compliance with Good Clinical Practice (GCP) and regulatory guidelines such as 21 CFR Part 11 via differentiated user roles and privileges, password and user authentication security, electronic signatures, SSL encryption, de-identification of Protected Health Information (PHI), and comprehensive auditing to record and monitor access and data changes. It also has a fully validated software development lifecycle (SDLC). These features distinguished the system over all other open source solutions for web based data collection we considered. There is an extensive feature list of the system located in the following URL if further information is required, https://www.openclinica.com/product-features.
The way in which the forms are designed is intended to be simple enough for non-technical people to design a system. To achieve this the platform uses a Microsoft Excel based spreadsheet model where the clinician/user simply adds data elements with some simple definitions of the data types and description etc, and this, when deployed, produces a set of web pages to capture each of these data items. Many examples of such forms will be shown throughout the rest of this document.
Key features of OpenClinica are:

- Open Source license
- Web based
- Supports all types of clinical studies
- No programming/ IT knowledge needed for CRF design
- Built on leading, independent standards
- Significant support for data validation and sign off

The main deficiency of OpenClinica is that, at the time of evaluation, it did not have the capacity or infrastructure components to manage large file uploads which on an imaging based project is a significant impairment. Since the evaluation of the system a community extension has been released for the system, from University of Aachen, called OC-Big which now supports large file transfers into the system (https://community.openclinica.com/extension/openclinica-big-data-oc-big). This system was only released in late December 2014 and, whilst very interesting, it is still quite immature. We have, therefore, decided not to deviate from our initial planning to use this, as the majority of the internal implementation work had already been done towards the solution described. We will, however, continue to monitor this development and should another requirement for prospective data capture present itself from within the project, or as a result of external engagement, we will revisit this decision since a single platform solution would be desirable.

2.2. XNAT EVALUATION

XNAT is primarily an image sharing and management portal. It was designed explicitly to support research and one can think of it as a PACS (Picture Archiving and Communication System) with extended components to facilitate the demands of clinical research. The features above those that we would expect in a clinical system are:

- Web based upload and download of DICOM studies (recent PACS systems support the WADO protocol now as an addition to the standard DICOM transfers but these are not used extensively at present).

- Segregation of images within the system e.g. image collections are separated by project and each project has its own set of users who can only see the relevant images. This kind of role based access to studies in a PACS system is not present in most systems.

- It supports access to the image file collections through protocols like WebDAV, which makes it simpler for researchers to access/use the data than the traditional DICOM interfaces which are very complex.

- Has an extended metadata model which is focussed on curation and validation, so checks of image quality and suitability for analysis tasks can be stored alongside the images in the database.

- Has the ability to store arbitrary file collections attached to the subjects record, and whilst these are not indexed in to the same level as the imaging studies this feature provides a useful addition for non standard file storage.

- It now has a feature to allow administrators to create data collection forms.

The last feature is where we found the system's main shortfall; in fact it does have the facility to extend the standard metadata model but this is very complex to implement and not easy to update. In addition it does not have some of the more advanced features of OpenClinica allowing record locking and sign off which are often key requirements in clinical trial scenarios. Also we did not find a way to implement data workflows in the system, i.e. users are presented with filtered lists of eCRFs for completion based on data already entered into the system. Finally whilst the interfaces for image access are based on well-established standards, the model for data collected into the forms engine is not and proved very difficult to produce a standardised solution for. A complete description of this issue can be found in Deliverable 3.1 as this problem needed to be addressed as part of the retrospective data provisioning. The solution presented there, whilst effective, is not one we would choose to pursue on a routine basis.

2.3. ARCHITECTURE SOLUTION

The two systems evaluated are excellent in their intended domains, and indeed both offer some functionality across the full solution space the project needs to cover, but neither would be sufficient in their own right as the final application stack. For this reason we decided to couple the two systems together to achieve a best of breed solution for the clinical users.

The task of coupling these two systems together was not trivial. Whilst they are both open source solutions, if we modified any of the code base, especially within OpenClinica, we would invalidate all the testing and certification the products have gone through. At the same time the workflow in a clinical trial data entry process is often such that the more common integration strategies of scheduled tasks checking for new data and updating another system are not viable. The solution needed has real time updates (or very close to it) in terms of data content but does not require any modification of the core products on either side. Fortunately there is a standard feature of OpenClinica that can be attached to the completion of any eCRF in the system and that is an email notification to someone (typically the study co-ordinator on subject recruitment or completion). It seemed possible to trap these triggered events from OpenClinica, use the content of the messages to interrogate the system and extract the data necessary to create an imaging "record" in XNAT which the researchers can then upload images into. This process will be elaborated on in more detail in section 0, but in summary it gives us the ability to slave the XNAT system to OpenClinica in near time to reduce the errors of managing data in two separate systems. We have called this component the OCX Integration Service. The high level structure of the system is shown in **Figure 1**.

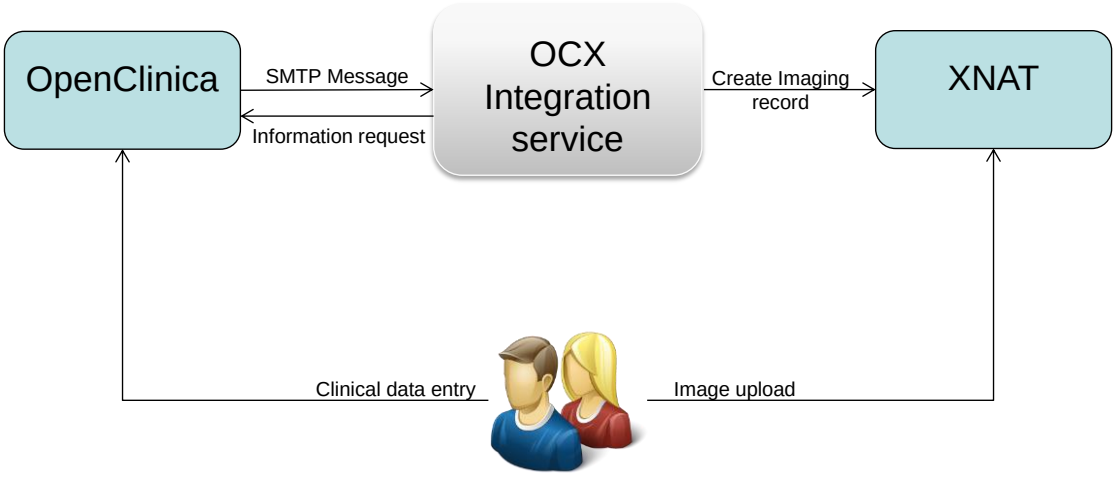


FIGURE 1: STRUCTURE OF DATA COLLECTION SYSTEM

3. SYSTEM IMPLEMENTATION

3.1. HOSTING

3.1.1. VMWare environment

The hosting for the data collection platform is on a virtualised environment based on the VMWare ESXi 5.5.0 product. This hypervisor is freely available, although not open source, and is installed on the following Hardware:

Model:	DELL PowerEdge R815
Processors:	32 CPU x 2.6 GHz
Processor Type:	AMD Opteron(TM) Processor 6212
Hyperthreading:	Inactive
Total Memory:	255.97 GB

Number of NICs:	4
Disks	
BigDisk:	Capacity: 10.91 TB (8.22 TB free)
datastore1:	Capacity: 128.50 GB (126.60 GB free)

The system has two disks; one is the internal 128GB capacity RAID1 mirrored operating system disk. The second is the external direct attached storage array which comprises 12 x 1.5TB disks configured in a RAID6 configuration across 10 disks with two hot spares.
This server is hosted in a server room in Sheffield Teaching Hospitals but connected to the University of Sheffield's network so it has a direct link to the UK academic high bandwidth network JANET which should ensure that hosting connectivity is not a bottle neck for large volume data transfers.

3.1.2. OpenClinica Server

This is a virtual machine running Ubuntu 14.01, and the virtual resources allocated to it are as follows:

Processors:	4 CPU x 2.6 GHz
Processor Type:	Virtual
Hyperthreading:	Inactive
Total Memory:	5.12 GB
Number of NICs:	1
Disks	
Disk1:	Capacity: 100GB

The OS installed is Ubuntu 64 bit 14.01 platform. This OS has also been patched for heartbleed and updated against any other bugs in the Linux system. 4 CPUs have been provisioned for it, with a total hard disk space of 100GB. OC was installed with Tomcat version 7.0.53 and Java7u65. The database engine used was PostgreSQL 9.3.5. This machine also has firewall exemption rules written to allow connections on port 443 and 22 only. A local Certificate Signing Request (CSR) was created using the standard tomcat keytool and a Terena SSL certificate was imported allowing secure connection to the OpenClinica site.

3.1.3. XNAT Server

This is a virtual machine running Ubuntu 14.04, and the virtual resources allocated to it are as follows:

Processors:	2 CPU x 2.6 GHz
Processor Type:	Virtual
Hyperthreading:	Inactive
Total Memory:	5.12 GB
Number of NICs:	1
Disks	
Disk1:	Capacity: 120GB

For this component we decided to use a standard virtual XNAT machine provided by the www.xnat.org site and changed all of the security settings. This server has a 64 bit Ubuntu 12.04 OS and was installed with Tomcat version 7.0.26 and Java7u51. Again this machine also has all the relevant updates and patches for security updates. The database engine used was PostgreSQL 9.1.14. The server also has firewall exemption rules opening only port 443 and 22 to clients. XNAT has the Apache web server sitting in front of Tomcat and is proxied using the standard jk_connector components that ship with apache. A CSR was created using OpenSSL allowing another Terena SSL certificate allowing for secure server site connections.
The images are stored off the server on a local NAS drive which currently has 12TB of storage configured as a single RAID6 volume with two hot spare disks (note this is not the same storage as is used by the VMWare environment for hosting the virtual servers described above).

3.1.4. Backups

The backups for these two servers are purely for the databases given that the actual imagestore for XNAT is not on the virtual machine itself anyway. Bash scripts were produced to run the backups nightly via a cron job and the backed up files are shipped off server to a local NAS on the local network. In addition these backups, located physically in the same room, are replicated to the central university data centre ~1km away from the operational servers.
In addition we keep a rolling backup on a nightly basis for 7 days so if necessary we can restore to any point in the last 7 days.
The images stored by XNAT are also backed up to this data centre, although it is not possible to keep rolling snapshots of a storage volume this big so it is simply a pure married backup of the file system that is replicated.
The only step we have not implemented yet is transaction log shipping off site so that in the event of a catastrophic database event we will be able to restore data back to within a few minutes of the failure as opposed to the previous night as it is the case now. The reason for this is primarily the cost of implementing such systems and maintaining them is often quite high and the risks of data loss are very low. This is due to the fact that all the clinical data in OpenClinica is captured to paper CRF in the clinical service and then transcribed into the database so there is always the source data available if it needs to be re-entered. This along with the fact that the recruitment rate in this project is not high means that the manual recovery procedure is far more resource efficient. The XNAT database can be reconstructed from the images so this is not a high risk either.
We are comfortable for the moment that the processes we have in place are commensurate with the risks we have identified but we will periodically review this situation and may choose to implement more sophisticated protocols in the future.

3.2. OCX INTERFACE IMPLEMENTATION

Here we will describe in a bit more detail the interface service produced to couple OpenClinica with XNAT. **Figure 2** shows the schematic overview of how the service is implemented. The service itself is written in Microsoft Dot.NET and runs under the mono (http://www.mono-project.com/) framework on Linux. In this instance the service is installed on the same server as OpenClinica but this is not a requirement and could run on an independent server if required.

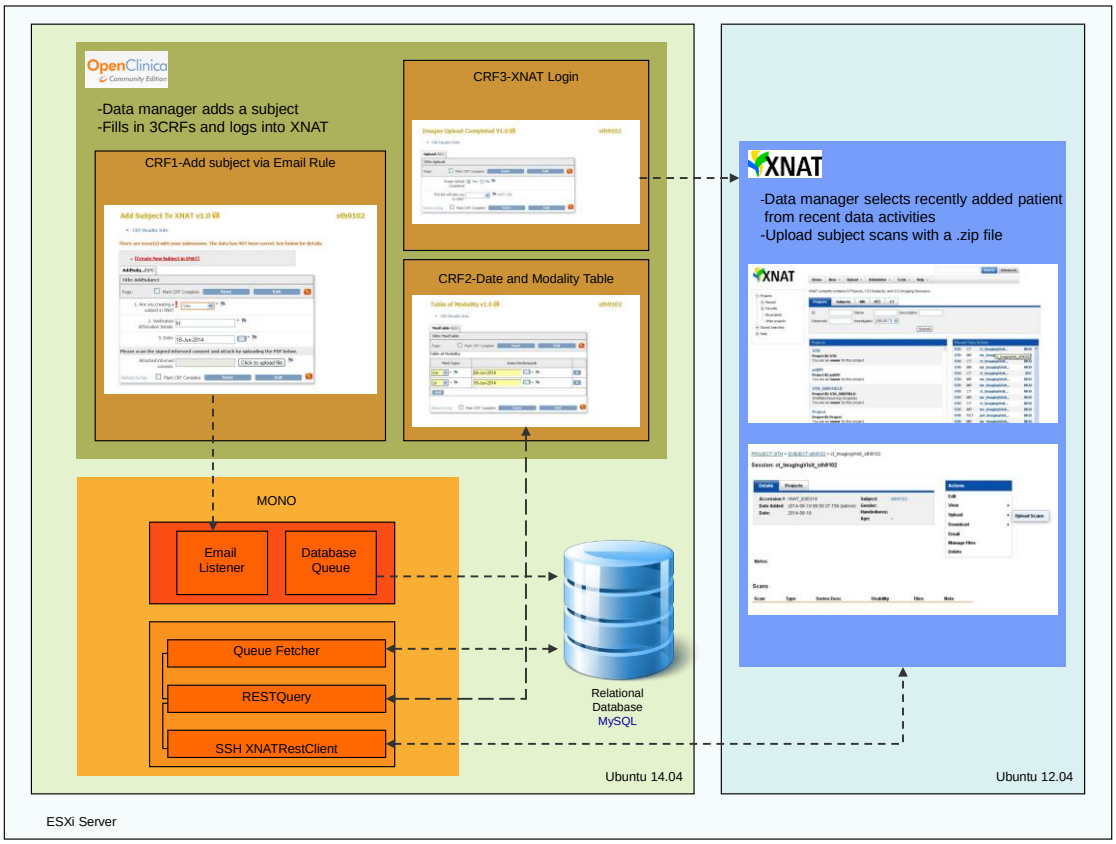


FIGURE 2: SCHEMATIC OVERVIEW OF OCX INTEGRATION SERVICE

The first stage of the OCX data integration service was to obtain the SMTP message using an email listener in MONO. This is then placed into a MySQL database queue. This database records the following:

- Incremental_number — Auto generated for processing
- study_name — From email
- sitename — From email
- subjectname — From email
- eventname — From email
- current_status — Current processing status of this transaction

Subject registration in XNAT only utilises the sitename (i.e. S_VPH) and subjectname (i.e. MAN88).
When a new SMTP message is received the relevant components of this message are parsed and placed into the database queue with a status of New. When the message processing thread next queries the queue, the oldest item with the New status will be

changed to processing. This state is where most of the OCX integration service work takes place.
In summary as shown in **Figure 3**, relevant data will be fetched from the database queue for the currently processing transaction. These, together with a further web services call to OpenClinica, will provide the information necessary to create the relevant experiment using an XNAT plug-in via SSH. These RESTful queries are used to obtain subject specific information. The database status queue will also then be updated to Completed once the XNAT plug-in returns a response confirming subject creation. These will be described in more detail in the next sections.

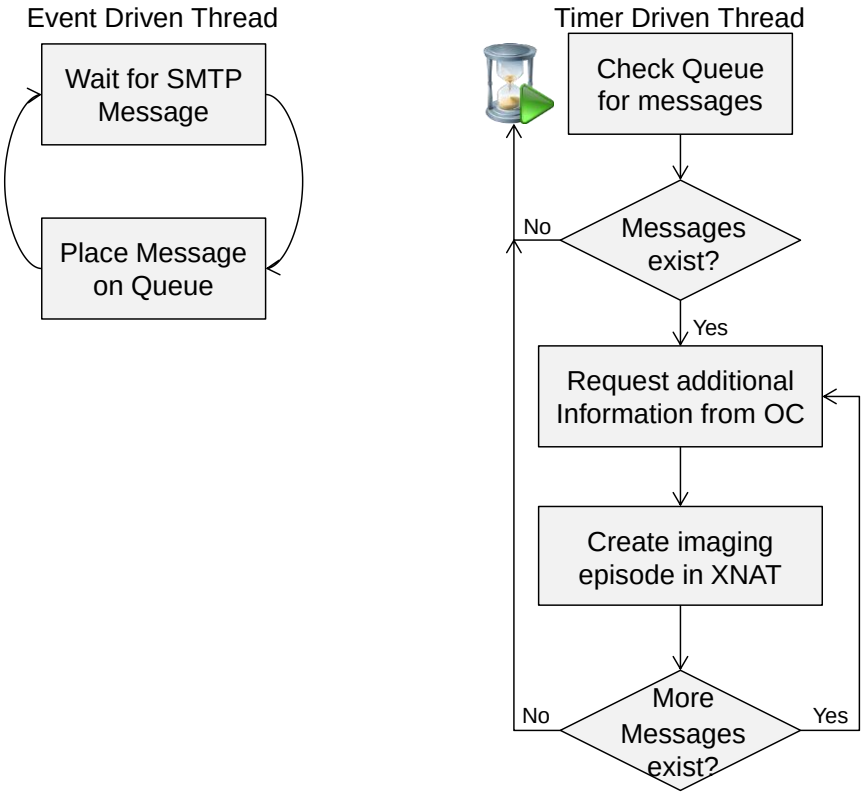


FIGURE 3: FLOWCHART FOR OCX INTERFACE

A detailed description of the webs services used for the interfacing of this service into OpenClinica is described in section 6.2 and the same code base was used in both systems. However there is one specific issue did affect the OCX service that was not relevant in the context of the research platform interfacing and this is the fact that OpenClinica modifies the internal subject ID from that specified by the user.

Whilst testing the OpenClinica and XNAT subject creation link, we discovered that OpenClinica only supports up to 8 unique characters. This could potentially be a problem as any repeating strings larger than 8 characters will trigger an automatically generated unique SubjectKey, sometimes referred to as key mangling. This is an issue because of the way in which the REST services rely on the SubjectKey to access data on a specific subject.

The core URL for obtaining all subject level data in the system has the following pattern:

```
https://{ServerName}/OpenClinica/rest/clinicaldata/xml/view/{Project}/{Sub
jectKey}/*/*
```

A specific example for the deployed system and test subject being:

```
https://vphdare-
oc.shef.ac.uk/OpenClinica/rest/clinicaldata/xml/view/S_VPH/SS_STHTEST/*/*
```

As an example we previously named one of our subjects STHTestBoy and then added a subject called STHTestBot shown by the XML excerpt in **Figure 4**. The subject STHTestBoy will have a SubjectKey of SS_STHTESTB whereas STHTestBot has a SubjectKey of SS_STHTESTB_7448. Here we can see how OpenClinica only takes 8 characters of the string STHTestBot and then randomly generates a 4 digit number after the 8 characters.
This behaviour needed to be managed reliably as OpenClinica only accepts REST queries based on SubjectKeys and not StudySubjectID as shown from the subject specific RESTful URL above. If we indeed used StudySubjectID the REST query would return nothing for the subject. Most partner sites will have subject IDs longer than the 8 unique character length restriction placed by OpenClinica.

```
<SubjectData SubjectKey="SS_STHTESTB_7448" OpenClinica:StudySubjectID="STHTestBot"
OpenClinica:Status="available" OpenClinica:SecondaryID="12344" OpenClinica:Sex="m"
OpenClinica:EnrollmentDate="2014-11-04">
          <StudyEventData StudyEventOID="SE_INITIALVISIT"
OpenClinica:StartDate="2014-11-04 00:01:00.0" OpenClinica:Status="data entry started"
StudyEventRepeatKey="1">
              <FormData FormOID="F_ADDSUBJECTTO_V10" OpenClinica:Version="v1.0"
OpenClinica:Status="initial data entry">
                  <ItemGroupData ItemGroupOID="IG_ADDSU_UNGROUPED"
ItemGroupRepeatKey="1" TransactionType="Insert">
                      <ItemData ItemOID="I_ADDSU_IMS_PDF" Value="" />
                      <ItemData ItemOID="I_ADDSU_IMS_DATE" Value="2014-11-05" />
                      <ItemData ItemOID="I_ADDSU_IMS_INIT" Value="kt" />
                      <ItemData ItemOID="I_ADDSU_IMS_CONF" Value="1" />
                  </ItemGroupData>
```

FIGURE 4: AN XML FRAGMENT SHOWING THE SUBJECT KEY MANGLING FOR IDS LONGER THAN 8 CHARACTERS

To resolve the issue of the 8 unique characters an additional REST query has to be made. A REST query call to for all subjects in the study is shown below:

```
https://vphdare-
oc.shef.ac.uk/OpenClinica/rest/clinicaldata/xml/view/S_VPH/*/*/*
```

Here we searched the StudySubjectID for the entire study (S_VPH) to retrieve their respective SubjectKey. After obtaining the correct SubjectKey we then used that information to obtain values from a CRF called Table of Modality. The event data specific XML is as shown below:

```
<FormData FormOID="F_TABLEOFMODAL_V10" OpenClinica:Version="v1.0"
OpenClinica:Status="initial data entry">
                  <ItemGroupData ItemGroupOID="IG_TABLE_MODALITYTABLE"
ItemGroupRepeatKey="1" TransactionType="Insert">
                      <ItemData ItemOID="I_TABLE_MODAL_DATE_TABLE" Value="2014-10-
01" />
                      <ItemData ItemOID="I_TABLE_MODAL_TYPE_TABLE" Value="1" />
                  </ItemGroupData>
              </FormData>
```

FIGURE 5: XML FRAGMENET SHOWING THE IMAGING STUDY INFORMATION

The ItemOID and Value of ItemData, as shown in **Figure 5**, are required. The value of I_TABLE_MODAL_TYPE_TABLE will be programmatically replaced to be a type of scan modality in this case "1" represents MRI scans. These fields will make the basis of experiment creation in XNAT via secure SSH.

This is done by using a plugin provided by XNAT called XNATRestClient over SSH to the XNAT virtual machine. This provides an easy way to call the XNAT REST API via the Java command-line tool. Some of the example calls that we used are as shown below:

```
XNATRestClient -host https://localhost -u username -p password -m PUT -
remote
"/data/archive/projects/VPH/subjects/man88/experiments/mr_030914_man88?xna
t:mrSessionData/date=03/09/14"
```

For this example a subject called man88 will be created with an experiment name called mr_030914_man88. The experiment convention is chosen to represent modality_date_subjectname to ease data entry in XNAT. Modality type and date was obtained from a REST query to CRF1 (**Figure 2:** Schematic overview of OCX Integration service) fields, whilst project name (VPH) and subject name/ID (man88) were obtained from the database queue. After creating a patient and the associated experiments we also did a verification check if this step has been successful. This is done before the database queue gets updated to Completed by using a call as shown below:

```
XNATRestClient -host https://localhost -u username -p password -m GET -
remote "/data/archive/projects/VPH/subjects/man88/experiments?format=csv"
```

If the subject exists in XNAT, the OSX integration service would flag the transaction as completed, otherwise it would set a Failed status in the queue.
Active monitoring of the queue is provided by the nagios system (http://www.nagios.org/), which checks the queue every three minutes and will email the systems administrators if any failed messages occur. These emails will then be actioned and the users contacted to ensure the case can progress unimpeded.

4. OpenClinica Data Entry and Configuration

Following the integration of OpenClinica and XNAT, we focused our attention on the creation of CRFs for clinical data entry. Data entry for most prospective data collection groups will fall into clinical CRF data entry and imaging data entry.
Using OpenClinica the clinical team will register the subject and fill in all the clinical CRFs ranging from subjects eligibility to neurological test results. The imaging team will then fill in the last 3 CRFs (of course in many places this might be the same team but it does not need to be).

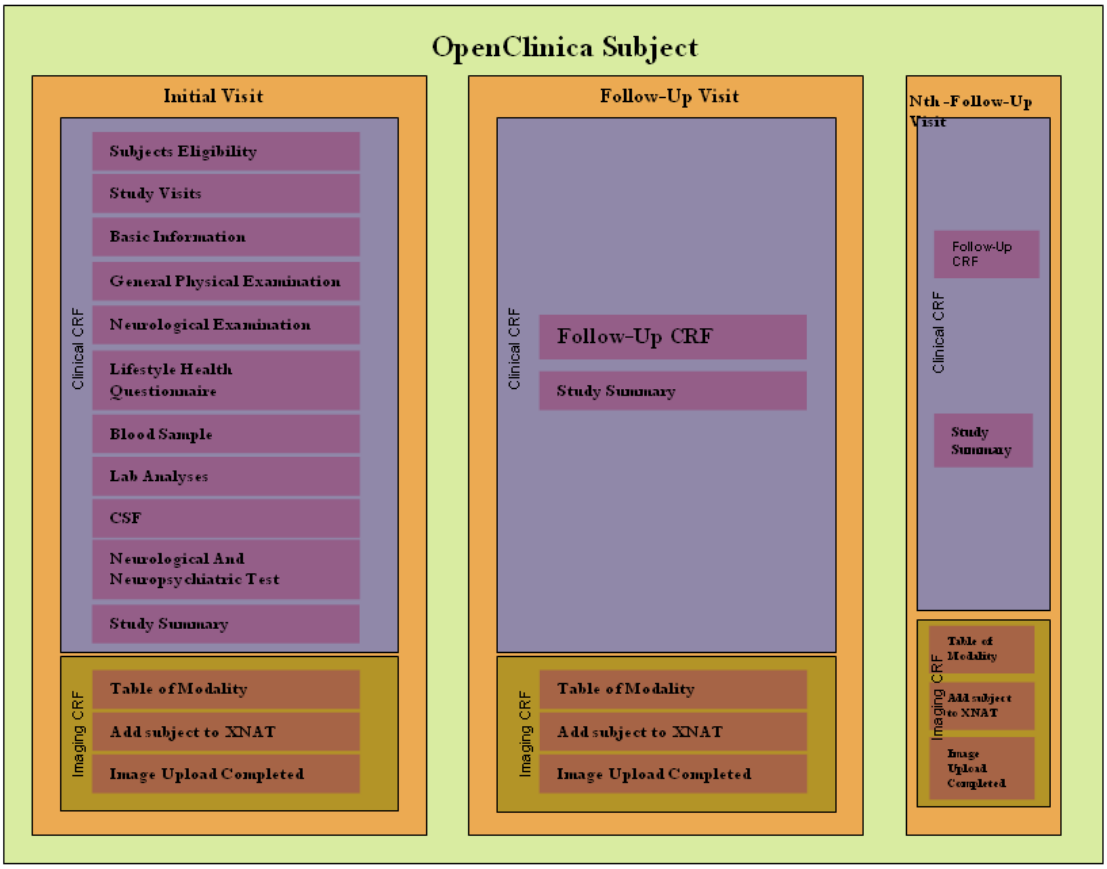


FIGURE 6: DIAGRAM SHOWING THE CRFS FOR A SUBJECT/VOLUNTEER FOR DIFFERENT VISITS

In the design we used a feature in OpenClinica allowing for repeating and non-repeating events. Here we set the initial visit to occur once (non-repeating) and set up the subsequent Follow-up visits (repeating) to be continuous. There are initially 14 CRFs to be completed, one of which is shown in **Figure 8**. These are comprised of 11 clinical based CRFs and 3 imaging CRFs. These are titled:

Clinical CRFs

- Subject Eligibility
- Study Visits
- Basic Information
- General Physical Examination
- Neurological Examination
- Lifestyle Health Questionnaire
- Blood Sample
- Lab Analyses
- CSF
- Neurological and Neuropsychiatric Test
- Study Summary

Imaging CRFs

- Table of Modality
- Add Subject to XNAT
- Image Upload Completed

Figure 7 below shows the OpenClinica user interface for subject data entry. After the registration of a subject-data entry, a person completes the first initial visit CRF. Then the imaging team will fill in the imaging CRFs and, as described in section Appendix 1, this will trigger the creation of the patient in XNAT.

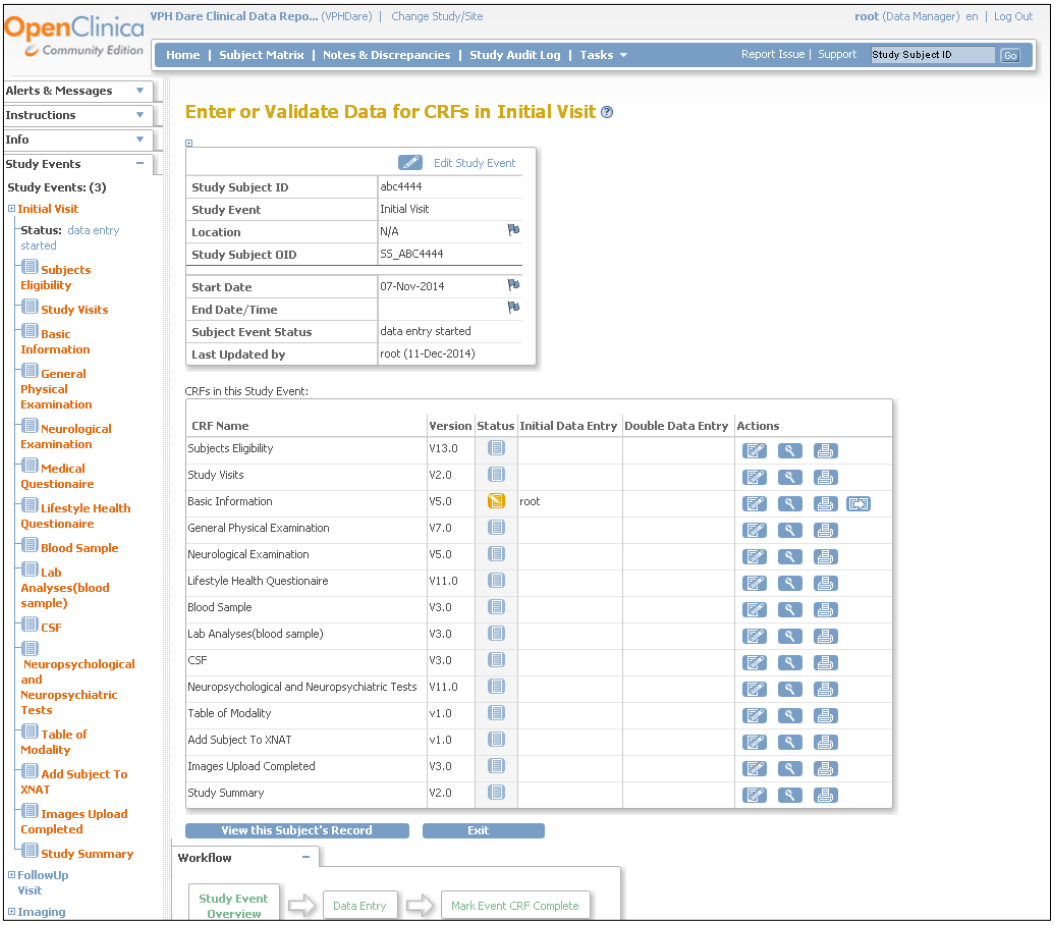


FIGURE 7: OPENCLINICA SUBJECT DATA ENTRY PAGE.

Once the subject is created the user then selects from the range of CRFs available which data entry tasks to perform. The subject level display summary shows the status of each CRF for this visit or event and who performed it.

The CRF data pages themselves have a simple layout and a typical example is shown in **Figure 8** this example contains a variety of input controls like radio button lists, drop down lists, dates etc.

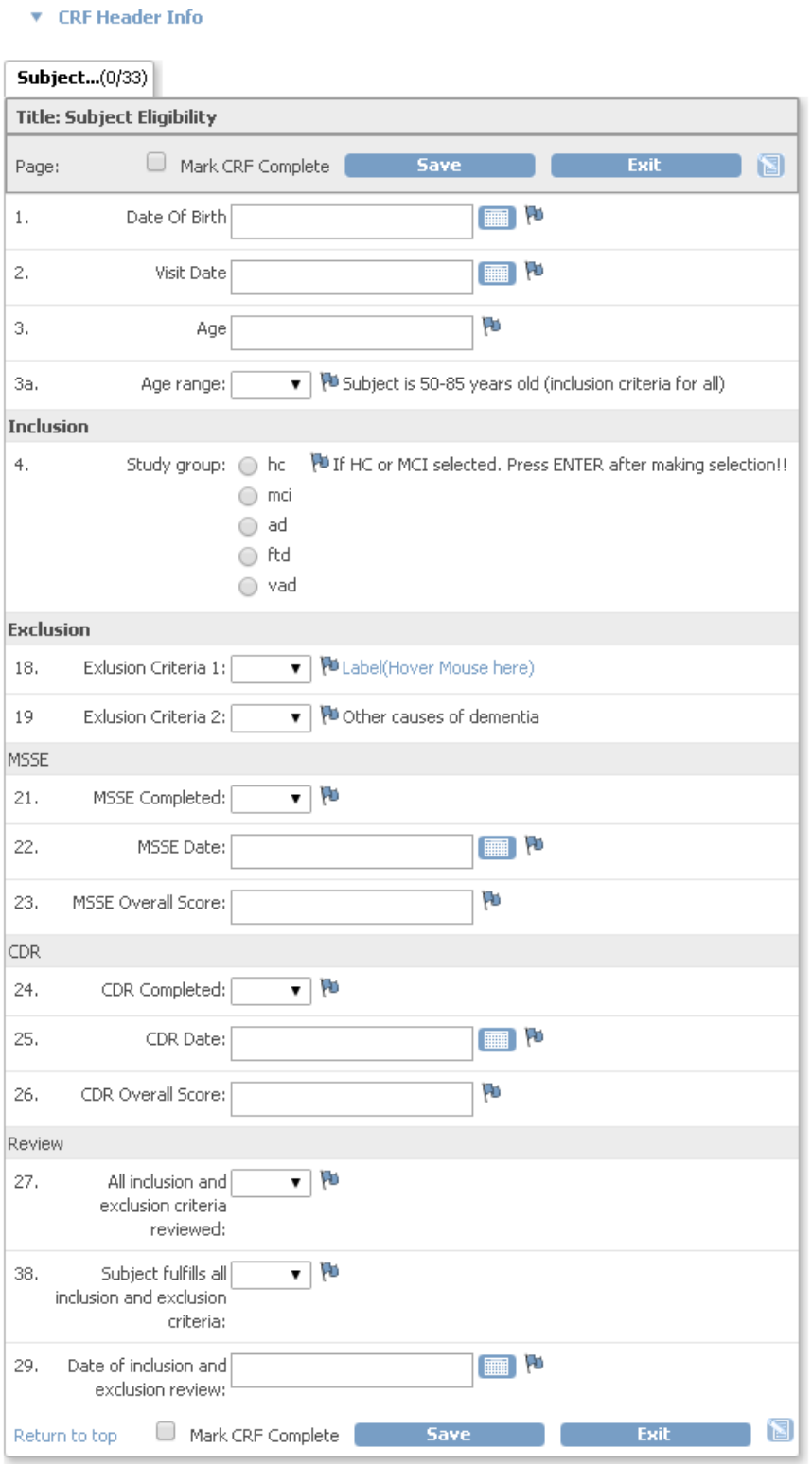


FIGURE 8: A TYPICAL CRF DATA ENTRY PAGE

This section is intended to give an overview of what has been delivered to the project, Appendix 1 has the full user manual provided to the clinical teams which contains a

very detailed view of the CRF's within the system and also the user view of uploading the images into XNAT.

5. XNAT DATA ENTRY AND CONFIGURATION

Since XNAT is essentially a sub-system to OpenClinica for user interactions and our configuration challenges are significantly less than those with the OpenClinica system, however there are still some constraints on its use.
The user interaction and interface is described in the prospective data entry manual Appendix 1 at the end of this document. We have chosen the upload format to be DICOM. This is because DICOM is a common format easily obtained from most imaging modalities, indeed the only format available on most clinical imaging modalities. DICOM also has a significant amount of metadata associated with it which makes it possible to perform automated validation of the uploaded images against the expected clinical records. Hence images will first be stored in XNAT prearchive before being archived fully in to the imaging repository. This is done so that images can be verified by a person before being fully committed into the database. This final validation step helps ensure the correct imaging has been "attached" to the correct patient since this is the most likely cause of data error in the collection system.

6. INTEGRATION WITH RESEARCH PLATFORM

The data publication suite (DPS) is the name of the software component used to "connect" to a data source, process the data in some way, annotate the data and then publish the results to an internet accessible server for consumption by the research community. The tool is designed to be used by scientists, as opposed to data managers, in the hope that they can take responsibility for managing their own data resources. There is also a significant advancement in this software over other tools which perform a simple ETL (Extract Transform Load) function, and that is the integration of semantic annotation into the published data framework. A detailed description of this software can be found in Deliverable 3.1 so it will not be repeated here but to give this development work some context we will describe the process for creating new data connectors and go onto the specifics of how this was implemented for the OpenClinica system.

6.1. REQUIREMENTS FOR A DATA SOURCE PLUGIN

Since any data source used in the platform must ultimately be converted to a relational database structure (MySQL) the nature of any plugin is to handle this transformation where necessary. To this end the two core requests a plugin must handle are:

- **GetSchema()** : Contains a list of tables, with their fields and data types plus any relationships that exist between them.
- **GetData (TableName)** : Returns a DataTable with the contents of the requested table. Each table is called sequentially until the whole set is produced.

In most sources this is a natural/trivial mapping onto the underlying data structures but in the case of OpenClinica this is not the case. A further desirable property is that the GetData method returns quickly as this will often be used extensively in the interactive development of the de-identification template.

6.2. OPENCLINICA INTERFACE

Fortunately the data model implemented in OpenClinica is based on a snowflake model, as are most automatically generated database applications, and this maps naturally onto the research platform hosting and query services (again these are defined in detail in D3.1). **Figure 9** shows the design of this model.

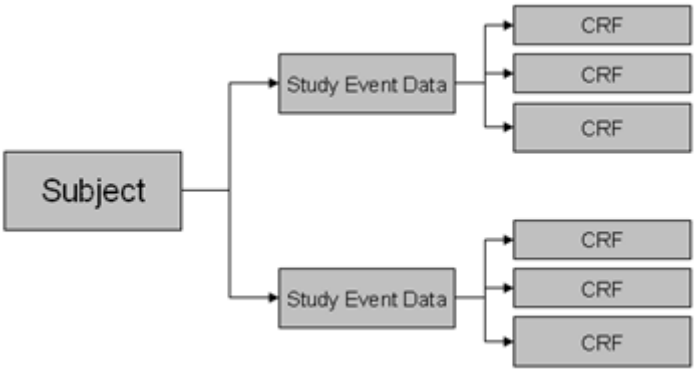


FIGURE 9: DATA MODEL USED BY OPENCLINCA

6.2.1. OpenClinica Web Service Integration

6.2.1.1. SOAP (Simple Object Access Protocol)

OpenClinica implements a SOAP web service interface which allows a significant amount of functionality to be executed without any user interaction. Experience in together data integration applications has shown us that given the choice between SOAP and REST (discussed in section 6.2.1.2) the development time and reliability of the former is much higher due to the WSDL (Web Services Description Language) definitions that are provided by SOAP interfaces. This allows a large amount of automated code generation to be provided to the developers and also gives a very robust way of updating and finding errors if the definitions from the third party system change for any reason.

Some of the key calls that are supported by OpenClinica are below and it was clear that for the interaction required for the project this would be more than sufficient.

- create (creates a new study subject)
- listAllByStudy (lists study subjects in a study)
- schedule (schedule a visit event)
- import (imports data)
- getMetadata (returns study metadata)

Due to the feature rich SOAP web service interface, this was our first approach to obtain CRF item data for a specific subject. We encountered issues using the OC SOAP service because the default mechanisms provided by the Dot.NET framework did not authenticate to the java spring security framework that OpenClinica uses. This is because there are no classes or methods built in natively supporting the .NET architecture supporting java spring web service security encryption (WSSE).

After further investigation, the correct login header information as shown in **Figure 10** is needed for OpenClinica SOAP authentication. We found out that the Dot.NET version used does not produce a SOAP authentication header which includes UsernameToken. One solution that we tried was to use a Dot.NET username token manager (WSE 3.0) to generate the missing UsernameToken. However this solution also produces additional information that is not needed by the java spring security check of the OpenClinica machine. The additional information is shown in **Figure 11** giving two additional fields Nonce and Created causing login authentication to fail.

```
<soapenv:Header>
 <wsse:Security  soapenv:mustUnderstand="1"  xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
 <wsse:UsernameToken                          wsu:Id="UsernameToken-27777511"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd">
 <wsse:Username>user</wsse:Username>
 <wsse:Password  Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-username-token-profile-1.0#PasswordText">password</wsse:Password>
 </wsse:UsernameToken>
 </wsse:Security>
</soapenv:Header>
```

FIGURE 10: CORRECT AUTHENTICATION HEADER REQUIRED BY OPENCLINICA

```
 <wsse:Nonce    EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-soap-message-security-1.0#Base64Binary"
>f8nUe3YupTU5ISdCy3X9Gg==</wsse:Nonce>
 <wsu:Created>2011-05-04T19:01:40.981Z</wsu:Created>
```

FIGURE 11: EXTRA INFORMATION SENT BY WSE3.0 .NET

Not being able to negotiate a secure connection to the SOAP web services essentially meant we could not use them for the project. This required an alternate approach discussed in the next section.

6.2.1.2. REST Web Services

Due to the incompatibility of the preferred OpenClinica SOAP solution with our Microsoft based development stack, we decided to pursue the REST solution instead. As described earlier we do not usually favour REST interfaces due to the lack of tooling to allow automated checking of interfaces etc. This is also compounded by the immaturity of the current implementation in OpenClinica which means we would expect to have significant refactoring work to perform if we move to subsequent versions of the product. However we do not anticipate an upgrade in the life of the project so the risk was deemed manageable.
The first issue we encountered was that the REST web services interface did not work before version 3.1.4. This required us to have to upgrade our OpenClinica version to 3.2.

In OpenClinica the REST web service is rather limited to filtering based on a RESTful URL. This will return either a json, html or XML format. A standard RESTful URL looks like:

```
http://localhost:8080/OpenClinica/rest/clinicaldata/
{format}/{mode}/{StudyOID}/{StudySubjectKey}/{StudyEventDefOID}[{StudyEvent
RepeatKey}]/{FormDefOID}
```

From the generic URL we can see that the user can specify output format (json, html or xml), mode (view or print) and also all the other subject specific information.

```
<soapenv:Header>
 <wsse:Security  soapenv:mustUnderstand="1"  xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
 <wsse:UsernameToken                          wsu:Id="UsernameToken-27777511"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd">
 <wsse:Username>user</wsse:Username>
 <wsse:Password  Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-username-token-profile-1.0#PasswordText">password</wsse:Password>
 </wsse:UsernameToken>
 </wsse:Security>
</soapenv:Header>
```

An example of a typical REST call to the OpenClinica services is the URL is shown below, this will return all subjects in a specific study, in this case with study OID called S_VPH. Note that a useful feature built into OpenClinica's REST interface is the use of * as a wildcard call.

```
https://vphdare-
oc.shef.ac.uk/OpenClinica/rest/clinicaldata/xml/view/S_VPH/*/*/*
```

This REST call is used to obtain all subjects in the study and all of their completed CRF records. A snippet of the folded XML output from the above call is as shown in Appendix 3.

6.2.2. OpenClinica data model

In the XML from Appendix 3 we see how the XML in OpenClinica is structured for a REST query to obtain all the data in a study. For this example the case study site is VPH (Study OID: S_VPH). OpenClinica generates its XML files based on the CDISC ODM standard (http://www.cdisc.org/odm) XML convention.
A more constructive description of the document format is provided in **Figure 12** and **Figure 13** which describes both the structure of the XML response document and also the way in which these elements map onto the underlying OpenClinica database schema, these images were copied from https://docs.openclinica.com/3.1/technical-documents/openclinica-3.1-database-model.

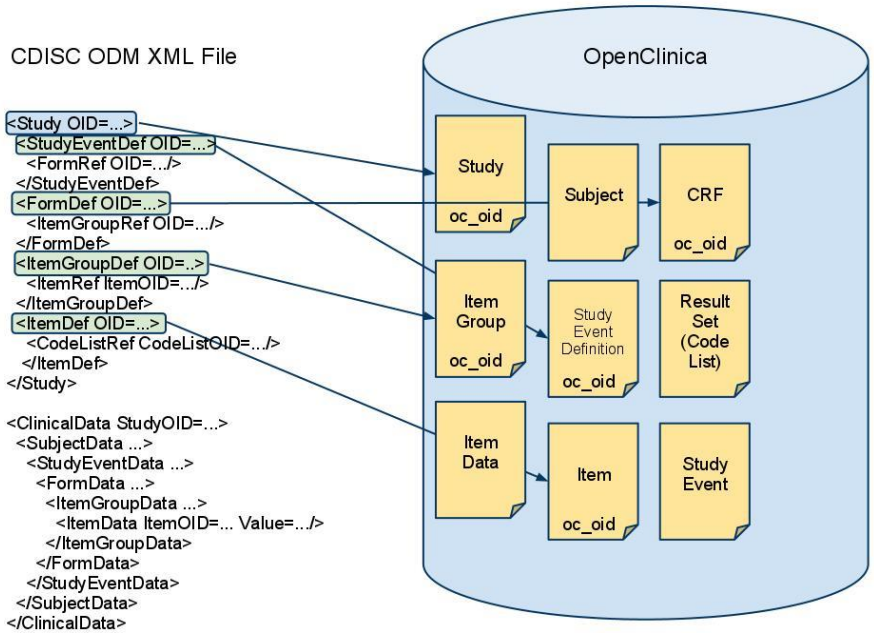


FIGURE 12: THE MAPPING MODEL FOR THE OPENCLINCAL METADATA SCHEMA ONTO THE UNDERLYING DATABASE SCHEMA

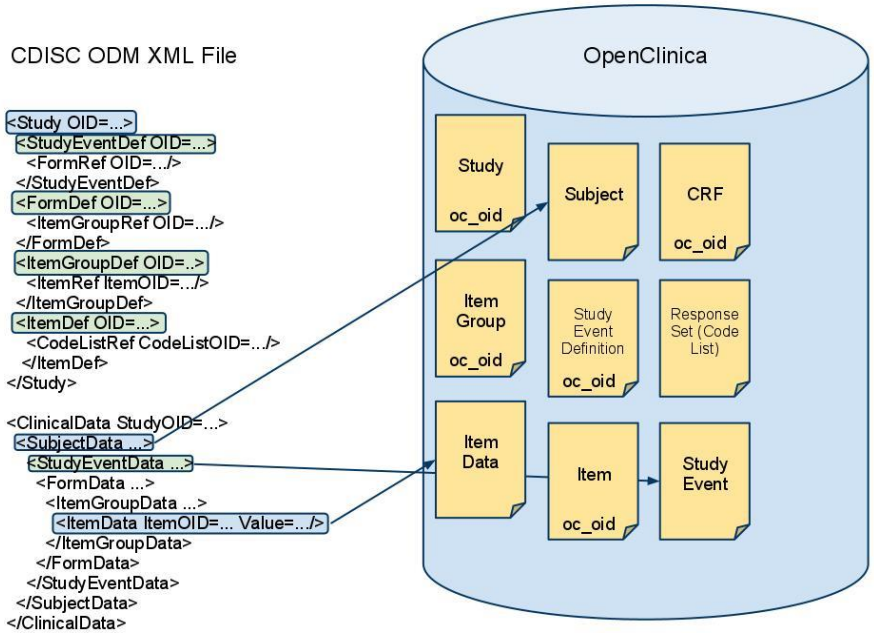


FIGURE 13: THE MAPPING MODEL FOR THE OPENCLINICA DATA SCHEMA ONTO THE UNDERLYING DATABASE SCHEMA

Crudely speaking the response is split into two parts, the schema definition containing information on the data groups, types and lookup value lists etc, along with the presentation state metadata such as the question text which appears next to the data entry box. The second block actually contains the clinical data entered where each data item is referenced back to the schema using a unique ID. This model in fact maps directly onto the functionality needed by the DPS during its ETL process. First it requests the data schema and then iterates through the data groups within it requesting the data.

6.3. DPS PLUGIN DEVELOPMENT

The plugin for connection to OpenClinica needs some core information regarding the location of the application, and it also needs login credentials to access those services. These are contained in the Connection group on the interface.
As with the XNAT browser we have made available a facility for browsing the data within the system as well. This feature is particularly useful when trying to visualise what the final data structures might look like when the export is completed. In this instance the user can decide which attributes of the data in OpenClinica should be used to generate the table and field names that will eventually be presented to the researchers. In this particular case the options are:

- OpenClinica ID
- Name
- OpenClinica ID & Name
- Comments
- Question

This may seem like an unusual list of options when you understand the internal workings of OpenClinica, but in fact where the meaningful information concerning a data item is held depends entirely on the person who designed the CRFs in the Excel template shown in Appendix 2. Having reviewed two other project configurations during the development of this code we decided to keep the options for naming as open as possible. The interface shows the "Create column names from" and "Create descriptions from" drop down controls which the users select. **Figure 14** shows the interface design.

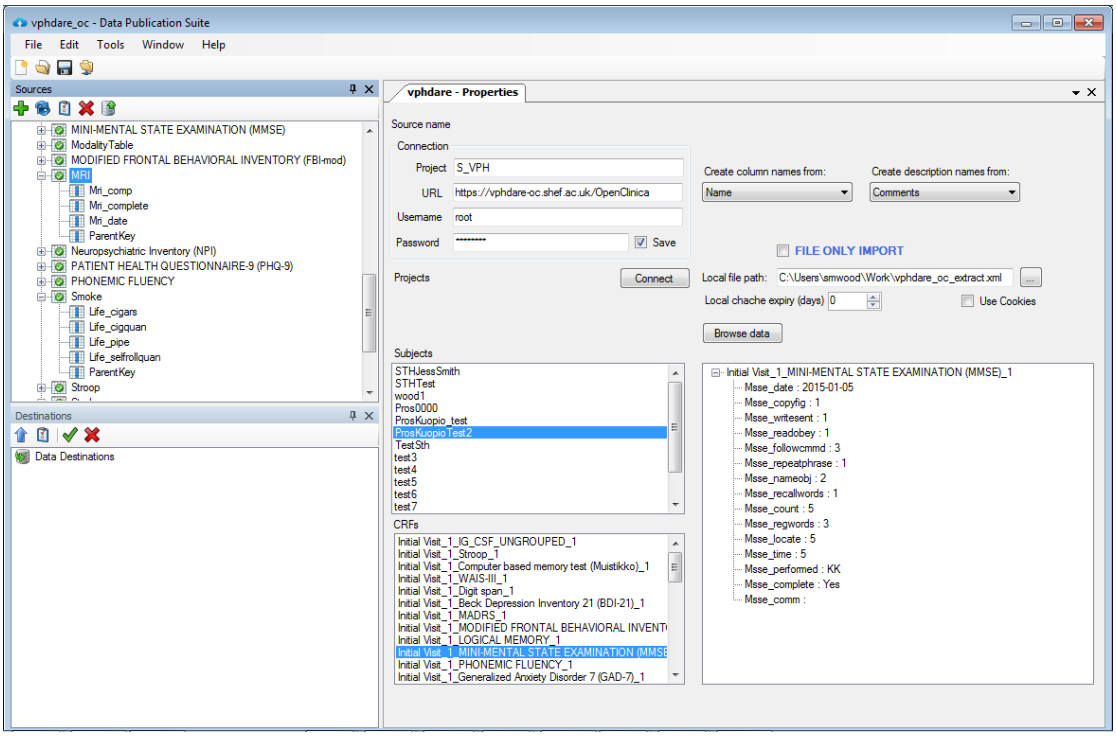


FIGURE 14: SCREENSHOT OF USER INTERFACE FOR DPS DATA EXTRACTION AND EXPLORATION

At any point the user can change these options and see the effect on the data structures that will be produced by browsing for subjects and their CRF data.
The final feature of the interface is that it will cache the patient data for a period of time. This is certainly desirable during the configuration of the system but will also ensure that the live server is only polled on a periodic basis no matter how frequently the extract is requested to run from within the DPS. This functionality is handled by saving the aggregated data document for all subjects and CRFs in a single XML file. One additional benefit of this feature is that we can now allow users to publish data from OpenClinica without ever having to have an account or connect live to the services because the format of the manual data export from the web page is exactly the same as that from the web service. This would allow an OpenClinica administrator to perform some predefined query on the system and pass it to the DPS without having to create special accounts etc, which is often quite desirable.

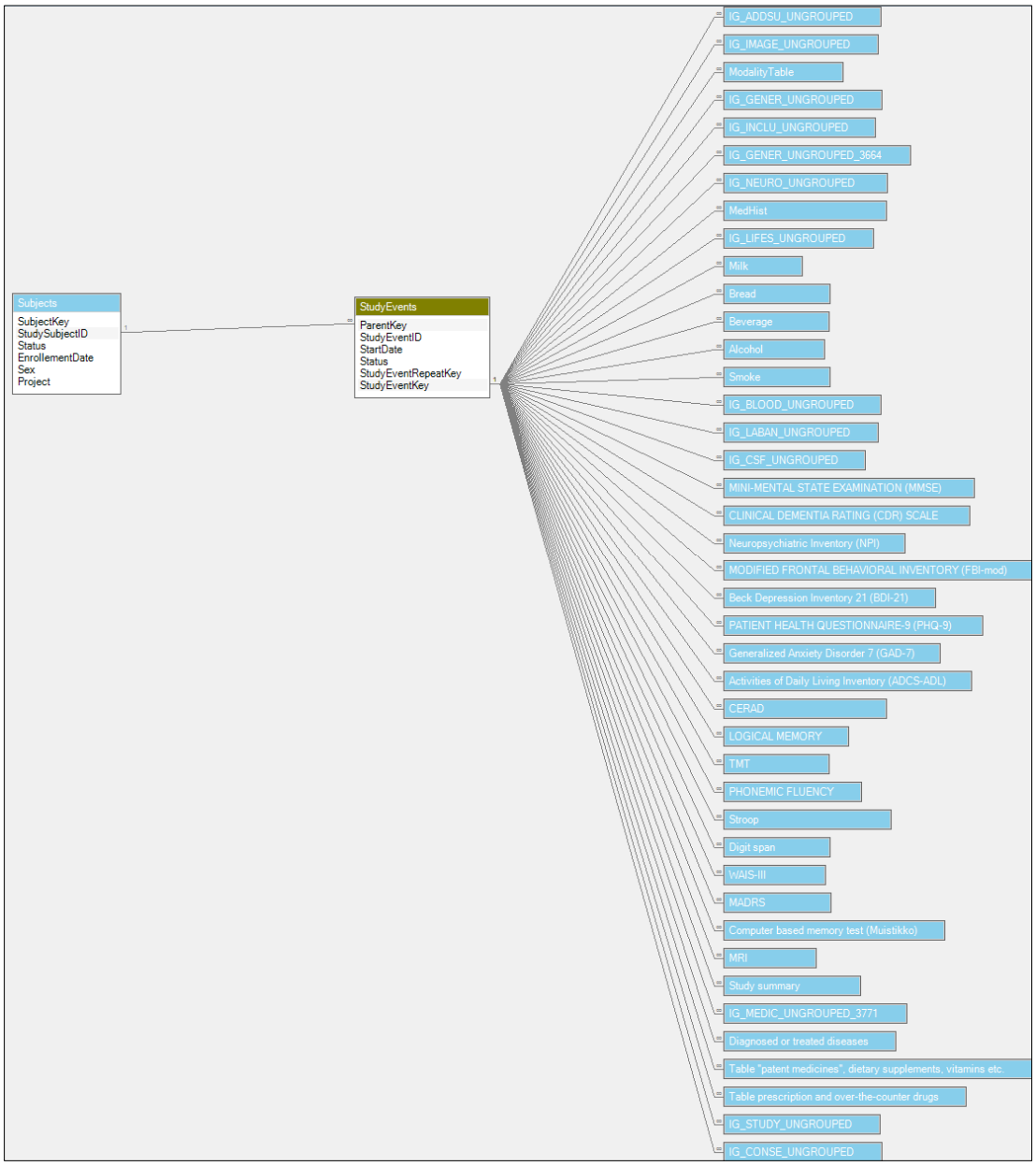


FIGURE 15: PROSPECTIVE TRIAL SCHEMA WHEN PUBLISHED

Following the schema analysis phase we are left with the relationship model shown in **Figure 15**. This is quite a lot larger than initially anticipated comprising of 47 individual tables, 789 individual data items. Whilst the size is not a problem technically, the concern is that with an information model this complicated the researchers who wish to find and use the data within the system will not be able to find it effectively.

At the point of writing there is not a significant amount of data entered into the electronic system as it is still undergoing the final iterations in terms of CRF design. However we do have several techniques for dealing with these issues; firstly, by default any tables or fields which are empty do not get published to the researchers and, whilst this should not be the case in a clinical trial, we do see sometimes that operational issues lead to some items being dropped once recruitment has reached full pace.

Secondly, and probably more valuable, is the feature of the DPS and research platform that allows only a subset of the source data to be published for any specific use. So we might imagine that there would be several different versions of this data containing different classes of data e.g. genetic, lifestyle or biochemistry. These datasets would have their own access policies and allow individuals to have a tailored "view" of the data to ease their exploration and usage.

7. NEXT STEPS

The platform built is very flexible and as is often the case we anticipate a number of developments to take place over the coming 12 months as people start to use it. In particular we expect that the XNAT system will need configuring to take more complex data types than the simple DICOM used for the prospective trials outlined in WP1. These developments will be taken on a case by case basis once representative data is available.

We will continue our open dialogues with WP1 and WP2 in order to meet their needs and do not anticipate any significant further development of this service, just small reconfigurations as the project requirements evolve.

APPENDIX 1 : USER MANUAL FOR PROSPECTIVE CLINICAL TRIAL

Introduction

This manual describes the process of registering a subject in OpenClinica and then uploading subsequent images via XNAT. Prospective imaging data collection is a two-step process. Firstly a subject has to be created in OpenClinica which will also host all the clinical data entry (CRFs). Next imaging data upload will be hosted in XNAT after the patient creation in OpenClinica.

Patient Creation and CRF input in OpenClinica

The site for OpenClinica is https://vphdare-oc.shef.ac.uk/OpenClinica . You should have already been sent credentials for accessing the system, if you have not please contact kevin.teh@sth.nhs.uk for an account.

Initial Visit

1. Login to the site using the username and password provided. This will take you to the welcome page as shown in Figure 16 below. On the menu bar click Tasks then under Submit Data click Add Subject.

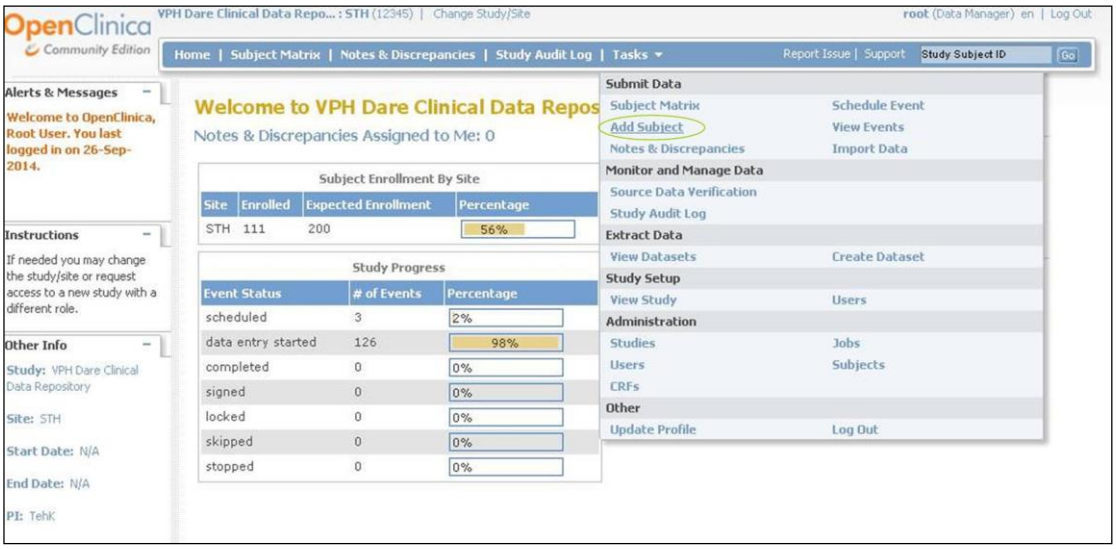


FIGURE 16: HOME SCREEN AFTER LOGIN

2. Fill in the Study Subject ID (it is very important to get this entry correct) the Sex of the subject and the date of enrolment in the study. Then click on Save and Assign Study Event as shown in Figure 17.

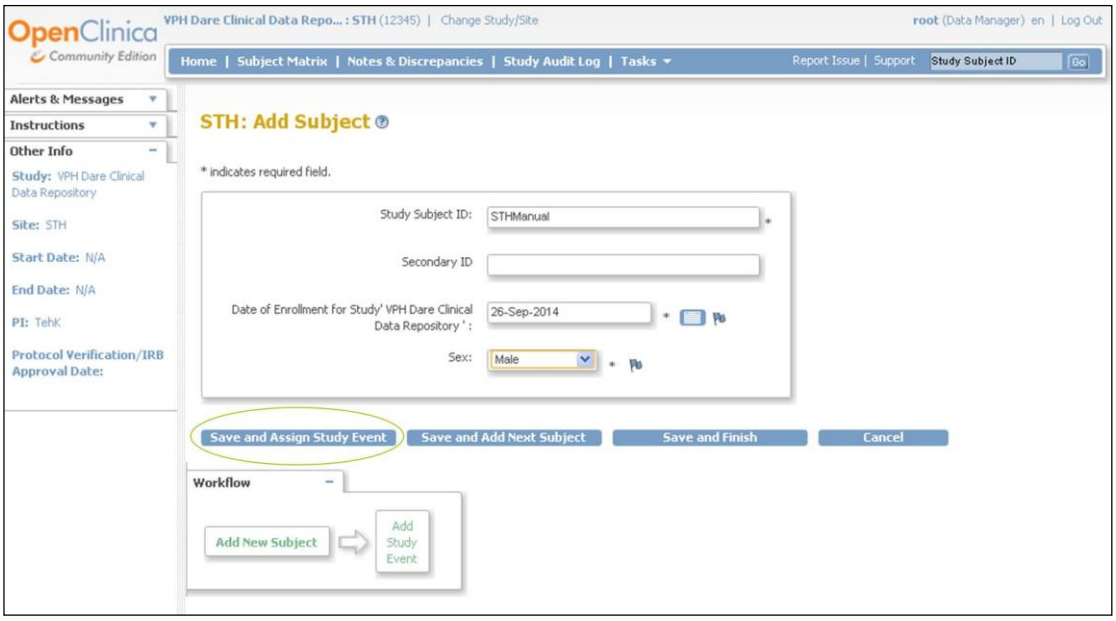


FIGURE 17: ADDING A SUBJECT

3. This will bring you to the Schedule study for subject page as shown in Figure 18. Study Event Definition should be set to Initial Visit (non-repeating). Set the Start Date/Time as the time of data entry (date and time now). Do not fill the field End Time as this is not required. Then click on Proceed to Enter Data.

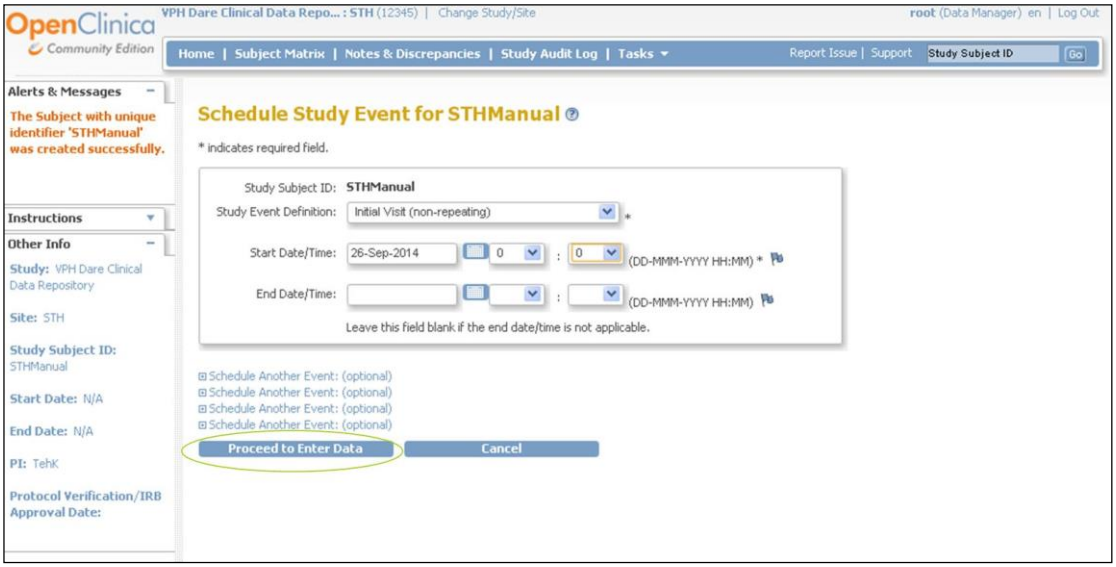


FIGURE 18: SCHEDULE STUDY EVENT

4. This will bring you to the **Enter or Validate Data for CRFs in Imaging Visit** (enter data page) page shown below in Figure 19. Now you have created a subject in OpenClinica.

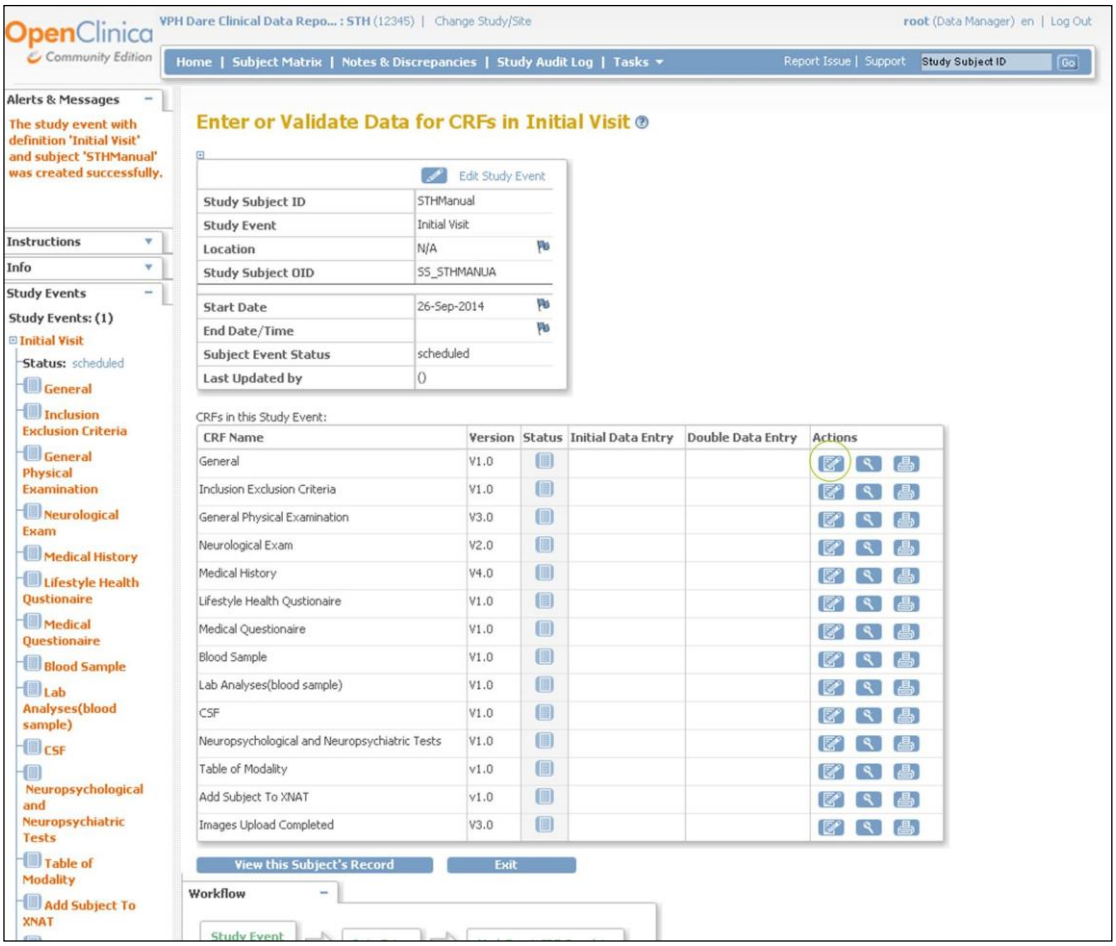


FIGURE 19: CRF DATA ENTRY PAGE

5. Fill in all the CRFs contained in the enter data page. This is done by clicking on the pencil (enter data) tab on the same row as the CRF name you want to fill under the Actions column. Most of them have validation rules associated with them and OpenClinica will throw up an error at the top of the CRF in red if (i.e.: a field was not filled). When a CRF is completed click Save. There is a Mark As Complete tab on top of the CRF next to the Save button. Only do this if you are sure all the data you have entered is correct. If you need to alter a field at any point after marking the CRF as complete you will have to fill in a discrepancy form by clicking on the flag next to the field you have just altered and explaining why you altered it.
6. After clicking Save you will be returned to the enter data page again. Continue doing this until you reach the last 3 CRFs (**Table of Modality, Add Subject To XNAT and Images Upload Completed**), these have to be **completed in sequence** to create a subject in XNAT.
7. Firstly complete the **Table of Modality** CRF. Here it is important to accurately enter firstly the modality (**Mod Type**) and then the **Date Performed** of that imaging modality as shown in Fig.5 below. If your subjects have had more than one imaging scan click the Add tab below the modality pull down menu and repeat the process again. Click Save when you have added the subject modalities and date scanned again shown in the Figure 20 example.

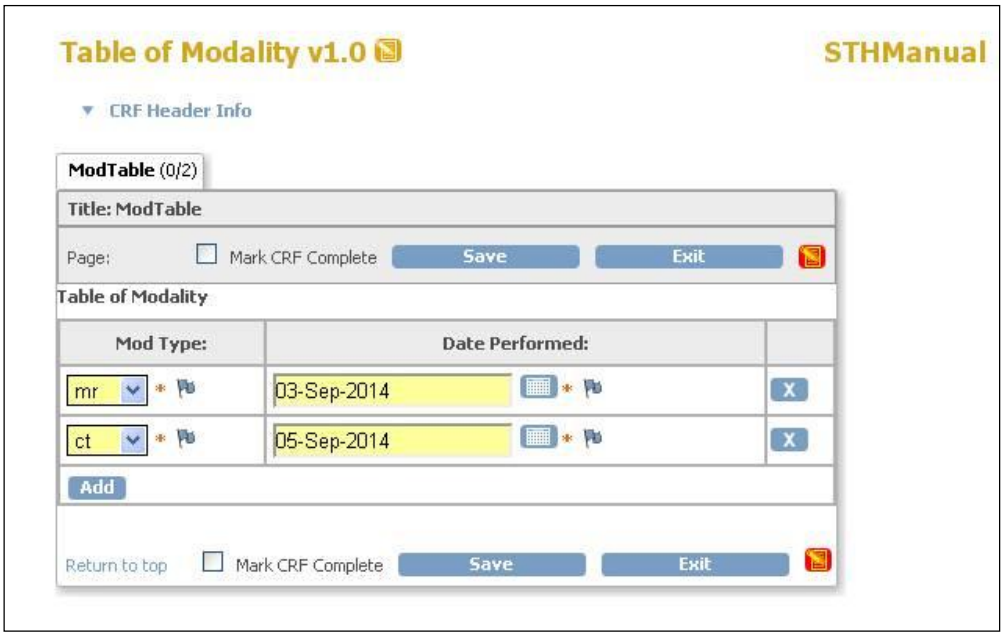


FIGURE 20: TABLE OF MODALITY CRF

8. This will return you to the enter data page. Now you will need to fill the **Add Subject To XNAT** CRF as shown in Figure 21. For question 1 select **Yes** then for question 2 key in your **initials** and then for question 3 enter the **date** (today). Also if you want you may also upload your subject informed consent. When you are done **click Save**. On the first Save click there will be a red warning prompting you to create a new subject in XNAT circled in Figure 21. On seeing this **click Save** again and you will again return to the enter data page.

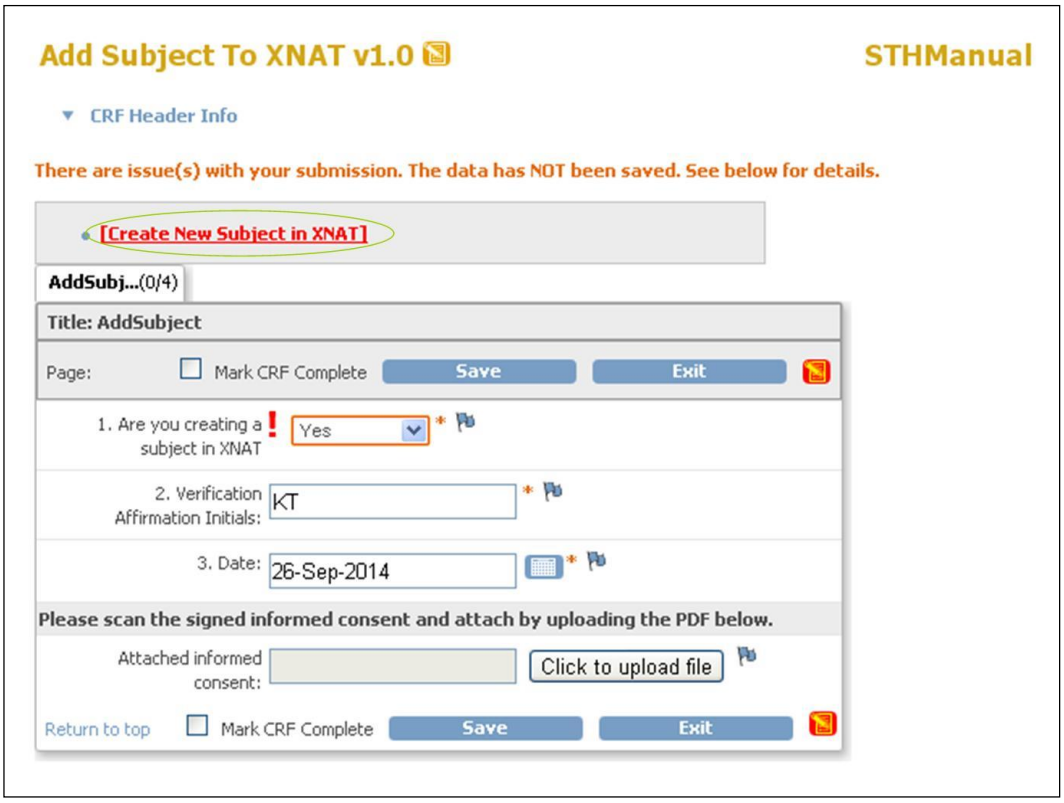


FIGURE 21: ADD SUBJECT TO XNAT CRF

9. Lastly fill in the **Images Upload Completed** CRF. Here there is a link that will take you to the XNAT site where the subject will have been created. It may take up to a couple of minutes for your new subject to be created. Click on the blue link called XNAT LINK and this will open a new tab with the XNAT login page. Next do the **XNAT** steps explained below and return to this step to verify that you have finished image upload by answering the two questions in this CRF.

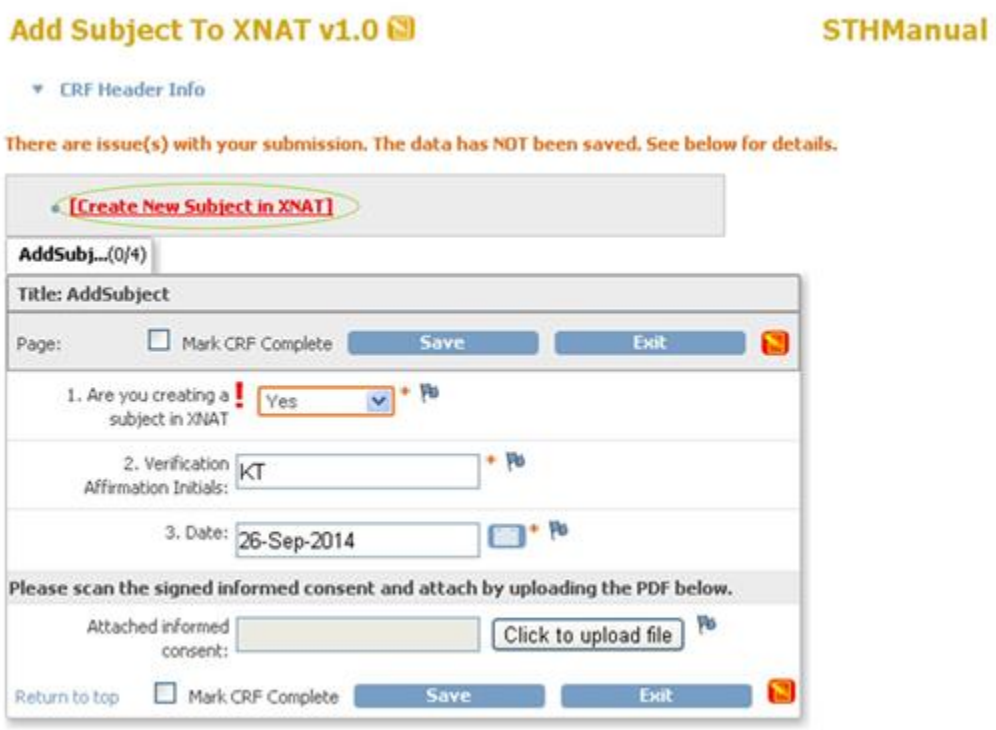


FIGURE 22: ADD SUBJECT TO XNAT CRF WITH HYPERLINK TO THE XNAT WEB PAGE

Follow-up Visits and Subsequent Follow-up Visits

1. Again login to OpenClinica with your credentials.
2. Next click on Subject matrix under Tasks and Submit Data as shown in Figure 23. This will take you to a page displaying the entire subject in this study site. Search for the patient you want to enter Follow-up data for. This is done either by scrolling through them or search for the name by typing keywords below the Study Subject ID tab.

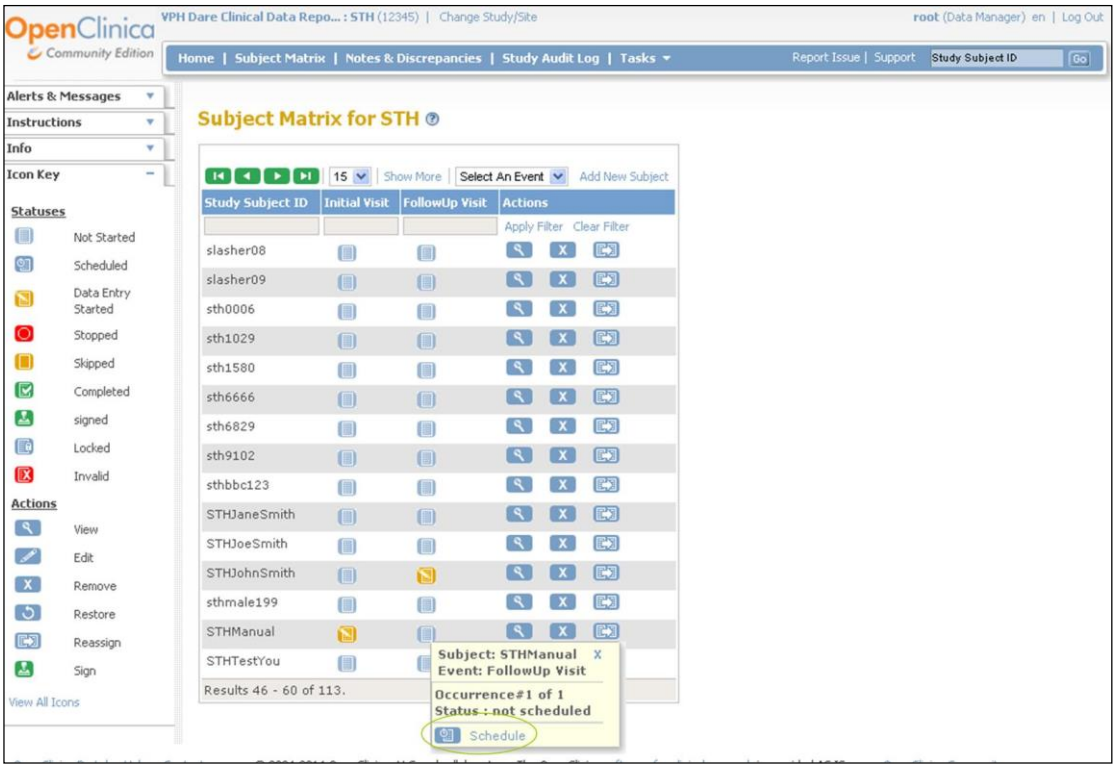


FIGURE 23: SUBJECT MATRIX FOLLOW-UP DATA ENTRY

3. Once you have found your subject, click the blue icon (Not Started) on the Follow-up Visit column then click schedule. Now do steps 3-9 from the ***Initial Visit*** steps above. One important difference is to setting Study Event Definition should be set to Follow-up Visit (non-repeating) in step 3 of ***Initial Visit*** instructions above.
4. **For subsequent Follow-up Visits** do steps 1-3 above but instead of a blue icon click on the yellow (Data Entry Started) icon then click on "Add Another Occurrence" as shown in Figure 24.

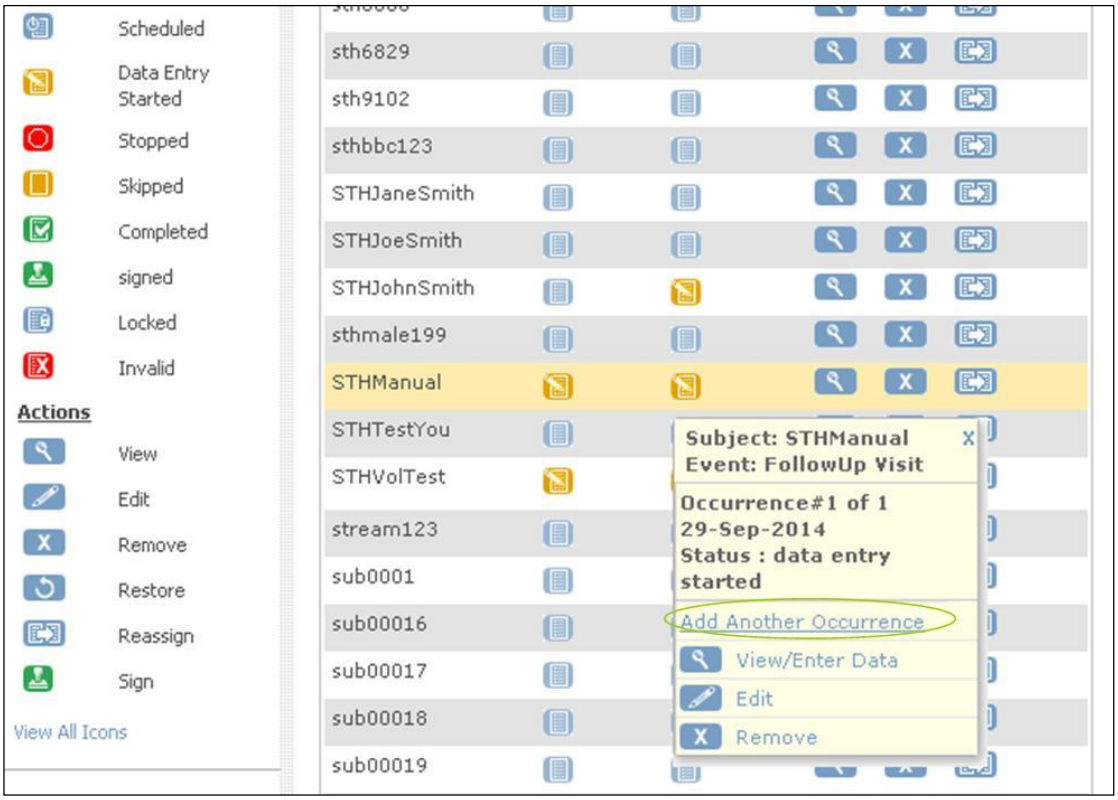


FIGURE 24: ADD ANOTHER OCCURRENCE FOR FOLLOW-UP VISIT

Also, if for whatever reason you were timed out (idle timeout 30mins), do steps 1 and 2 as described by the ***Follow-up Visits and Subsequent Follow-up Visits*** instructions above. And to continue where you were, select and click on the appropriate visit column and then click on View/Enter Data.

OpenClinica Quirks

CRF-Subjects Eligibility

After keying in date of birth and visit date press TAB/CTRL to automatically calculate the age and fill the Age field.

If someone intends to enter a multi-select value as shown in Figure 25 below, hold ctrl and click on the next option.

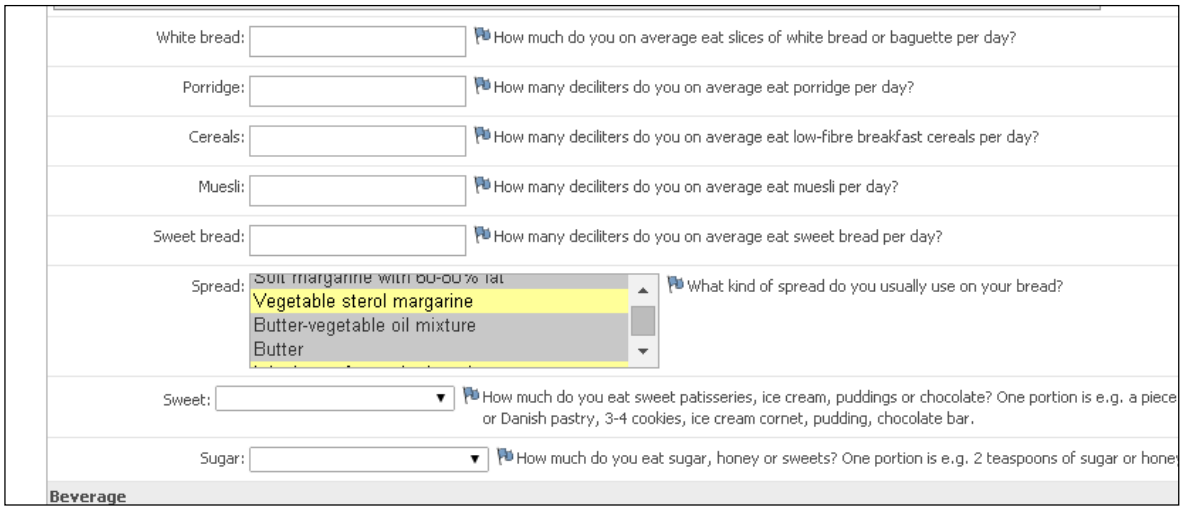


FIGURE 25: MULTI-SELECT EXAMPLE

Also for any calculations/score (i.e. Neuropsychological and Neuropsychiatric Tests CRF) the scores/results will only be calculated once the CRF is saved.

Medical History

For partial dates key in manually (i.e.: Feb-2015) for onset and start time in the medical History and Medications table respectively.

Neuropsychiatric Test NPI

Total scores will only be calculated when you press ENTER or SAVE the CRF. If SAVED to view it go back into the CRF again.

Image Upload in XNAT

The site for XNAT is https://vphdare-xnat.shef.ac.uk . In order to upload an imaging study the only acceptable image is a DICOM **study folder** compressed in a **.zip** file. This is easy to achieve and if you do not have a client already installed a good open source application for this is 7-Zip which can be downloaded from:

http://www.7-zip.org/

1. Login into XNAT using the username and password provided.
2. On the Recent Data Activity Tab you should be able to see your newly created subject in OpenClinica. The subject label created for your convenience is split into {modality_date }_{d/m/yr}_{subjected} (created in OpenClinica) (i.e.: **mr_030914_STHTestBoy**). Hover over the label to get more information if only partially displayed. Firstly **click** on the **subject label** as shown in Figure 26.

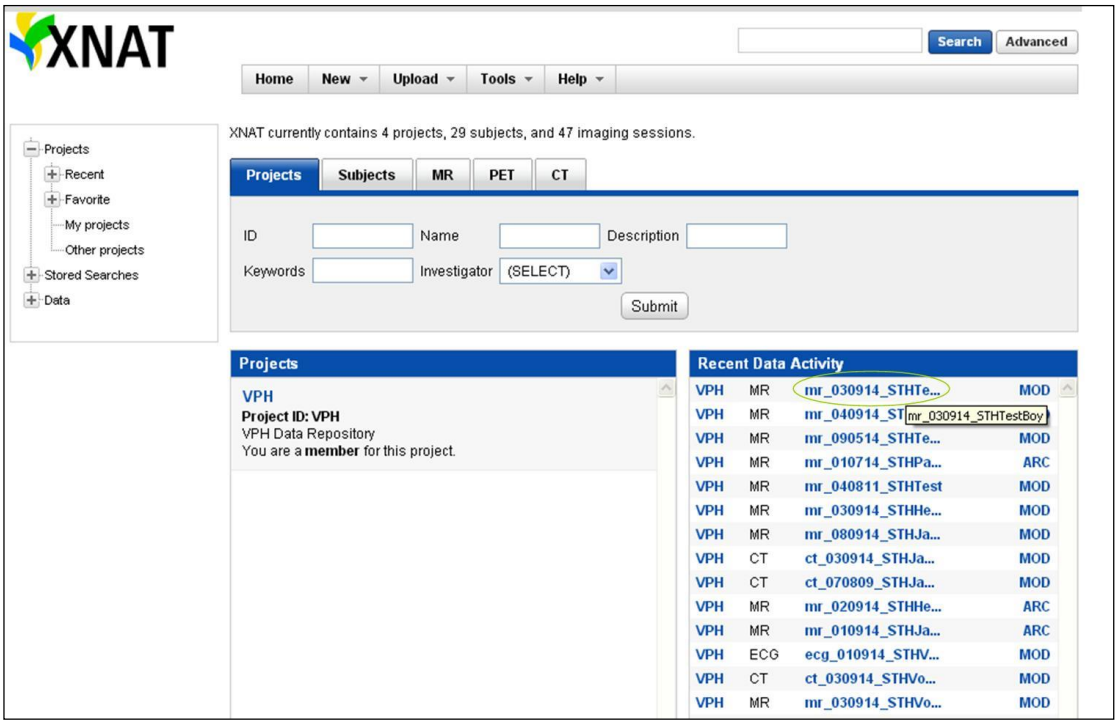


FIGURE 26: XNAT HOMEPAGE SELECT PATIENT TO ADD IMAGING SCANS

3. This will take you to the page for **Session mr_030914_STHTestBoy mr_301014_testsub**. Click on **Upload** then upload scans under Actions this is shown in Figure 27.

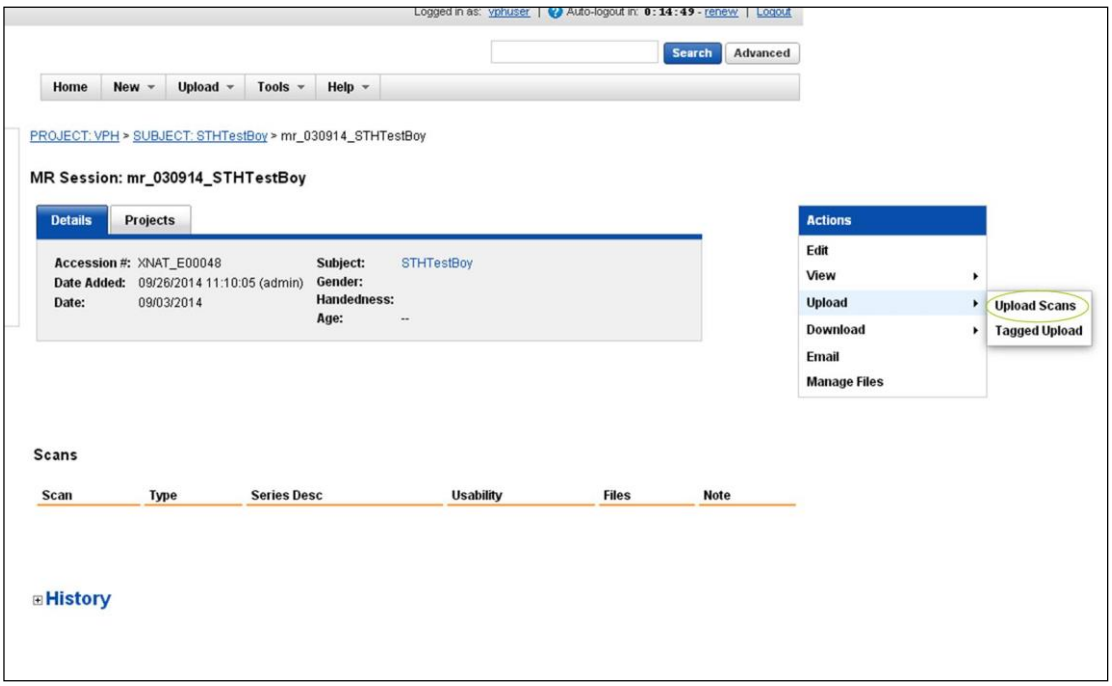


FIGURE 27: XNAT UPLOAD SCANS

4. After clicking Upload Scans, a compressed upload screen will pop-up. Click on the **Prearchive** radio button then click Choose File as shown in Figure 28. Now select the imaging files that you want to store and click Upload. When

completed it will give you a Successfully Uploaded 1 Session into the Prearchive message, below the upload status bar when it is completed as shown in Figure 29.

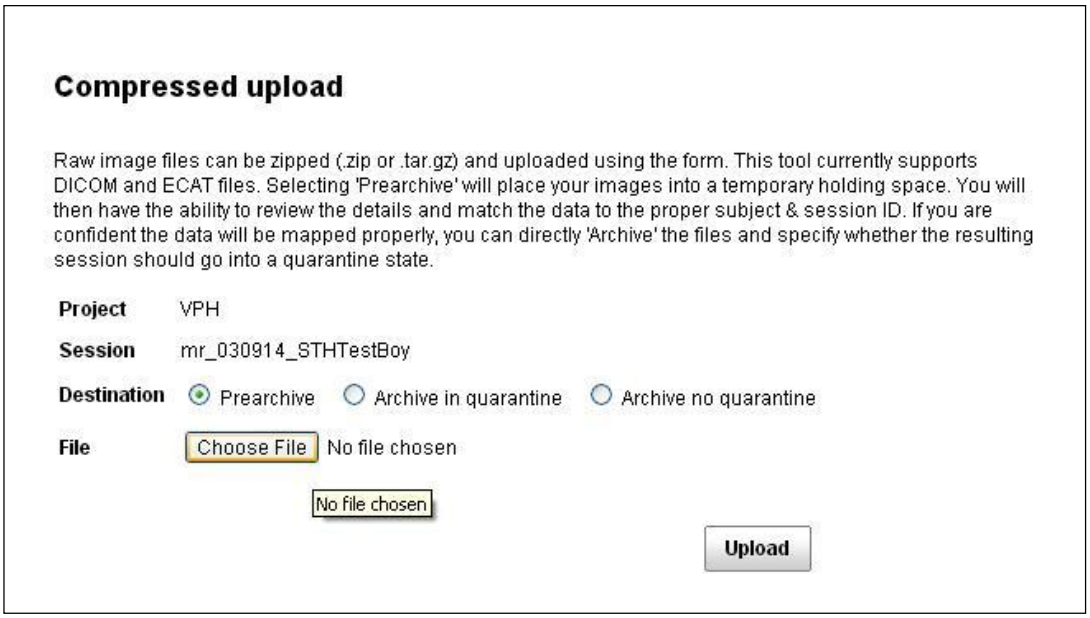


FIGURE 28: FILE UPLOAD POP-UP PAGE

5. Lastly return to step 7 for Patient entry in OpenClinica and complete the **Images Upload Completed** CRF.

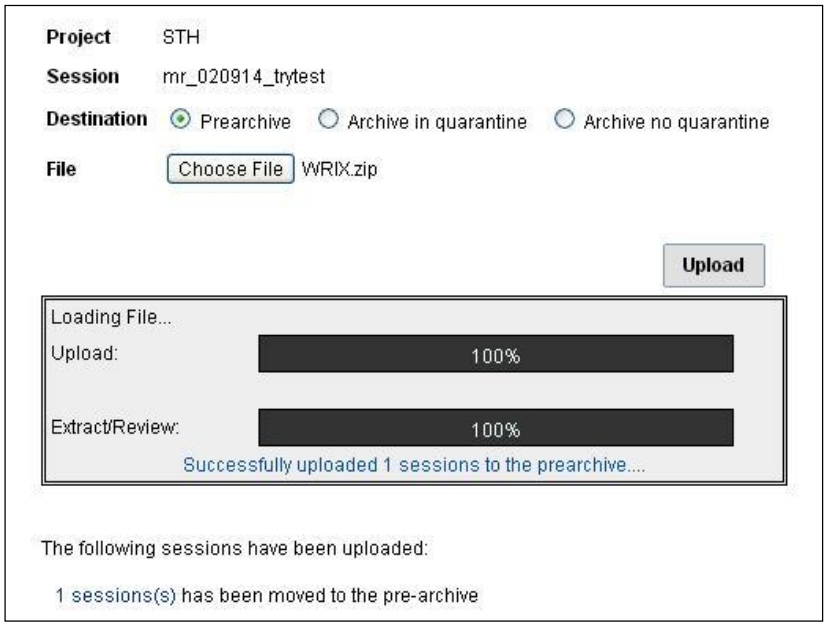
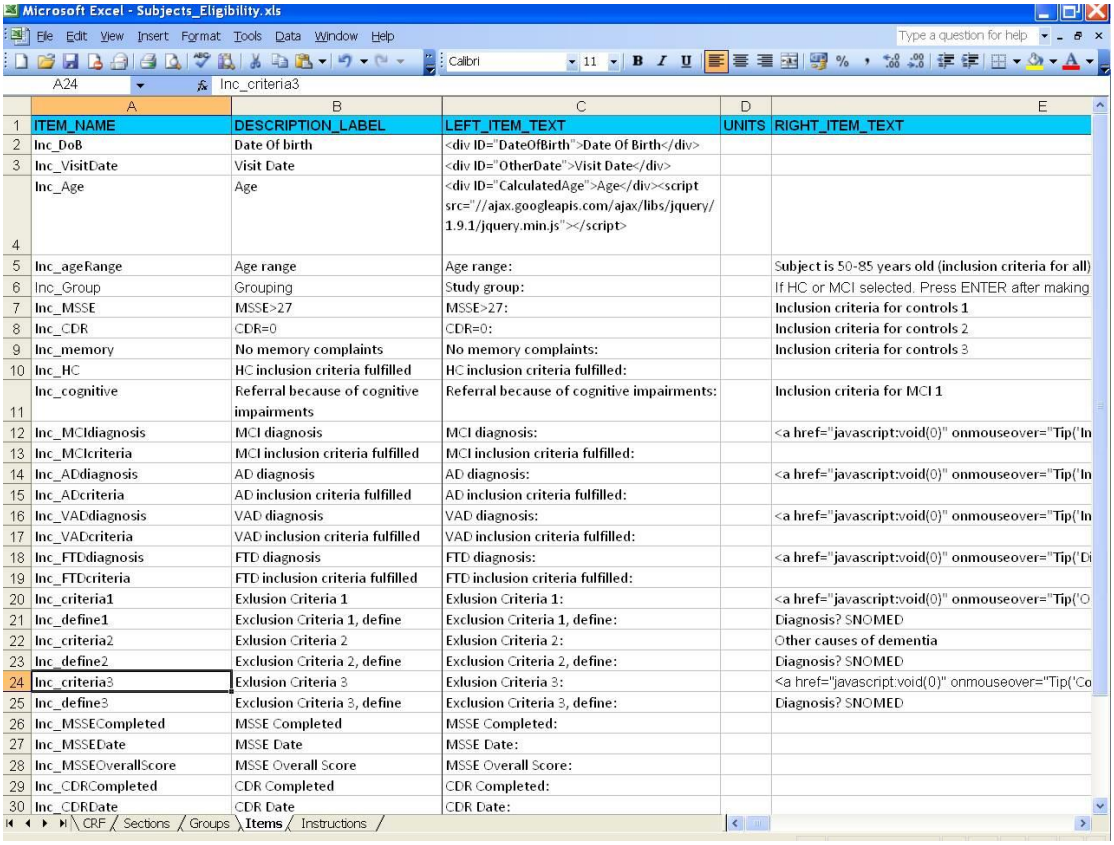


FIGURE 29: FILE UPLOAD COMPLETED

Housekeeping

If you lose your OpenClinica or XNAT passwords email kevin.teh@sth.nhs.uk for a new password.

APPENDIX 2 : EXCEL SPREADSHEET REPRESENTATION OF A CRF



Microsoft Excel - Subjects_Eligibility.xls

File Edit View Insert Format Tools Data Window Help

A24 Inc_criteria3

	A	B	C	D	E
1	ITEM_NAME	DESCRIPTION_LABEL	LEFT_ITEM_TEXT	UNITS	RIGHT_ITEM_TEXT
2	Inc_DoB	Date Of birth	Date Of Birth		
3	Inc_VisitDate	Visit Date	Visit Date		
4	Inc_Age	Age	Age <script src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script>		
5	Inc_ageRange	Age range	Age range:		Subject is 50-85 years old (inclusion criteria for all)
6	Inc_Group	Grouping	Study group:		If HC or MCI selected. Press ENTER after making
7	Inc_MSSE	MSSE>27	MSSE>27:		Inclusion criteria for controls 1
8	Inc_CDR	CDR=0	CDR=0:		Inclusion criteria for controls 2
9	Inc_memory	No memory complaints	No memory complaints:		Inclusion criteria for controls 3
10	Inc_HC	HC inclusion criteria fulfilled	HC inclusion criteria fulfilled:		
11	Inc_cognitive	Referral because of cognitive impairments	Referral because of cognitive impairments:		Inclusion criteria for MCI 1
12	Inc_MCIdiagnosis	MCI diagnosis	MCI diagnosis:		<a href="javascript:void(0)" onmouseover="Tip('In
13	Inc_MCIcriteria	MCI inclusion criteria fulfilled	MCI inclusion criteria fulfilled:		
14	Inc_ADdiagnosis	AD diagnosis	AD diagnosis:		<a href="javascript:void(0)" onmouseover="Tip('In
15	Inc_ADcriteria	AD inclusion criteria fulfilled	AD inclusion criteria fulfilled:		
16	Inc_VADdiagnosis	VAD diagnosis	VAD diagnosis:		<a href="javascript:void(0)" onmouseover="Tip('In
17	Inc_VADcriteria	VAD inclusion criteria fulfilled	VAD inclusion criteria fulfilled:		
18	Inc_FTDdiagnosis	FTD diagnosis	FTD diagnosis:		<a href="javascript:void(0)" onmouseover="Tip('Di
19	Inc_FTDcriteria	FTD inclusion criteria fulfilled	FTD inclusion criteria fulfilled:		
20	Inc_criteria1	Exlusion Criteria 1	Exlusion Criteria 1:		<a href="javascript:void(0)" onmouseover="Tip('O
21	Inc_define1	Exclusion Criteria 1, define	Exclusion Criteria 1, define:		Diagnosis? SNOMED
22	Inc_criteria2	Exlusion Criteria 2	Exlusion Criteria 2:		Other causes of dementia
23	Inc_define2	Exclusion Criteria 2, define	Exclusion Criteria 2, define:		Diagnosis? SNOMED
24	Inc_criteria3	Exlusion Criteria 3	Exlusion Criteria 3:		<a href="javascript:void(0)" onmouseover="Tip('Co
25	Inc_define3	Exclusion Criteria 3, define	Exclusion Criteria 3, define:		Diagnosis? SNOMED
26	Inc_MSSECompleted	MSSE Completed	MSSE Completed:		
27	Inc_MSSEDate	MSSE Date	MSSE Date:		
28	Inc_MSSEOverallScore	MSSE Overall Score	MSSE Overall Score:		
29	Inc_CDRCompleted	CDR Completed	CDR Completed:		
30	Inc_CDRDate	CDR Date	CDR Date:		

CRF / Sections / Groups / Items / Instructions /

FIGURE 30: CRF EXPORT TO SPREADSHEET

The CRF platform is both feature rich and simple. To hide or show items is as simple as setting conditional flags on a excel column. We have also explored using javascript for calculation, to hide long strings of texts/labels using the onmouseover tip and untip javascript function and also putting the XNAT link using html tag.

APPENDIX 3 : EXAMPLE OF CDISC ODM DATA DOCUMENT

```
<?xml version="1.0" encoding="utf-8"?>
<ODM    FileOID="Study-MetaD20141111135818+0000"    Description="Study    Metadata"    CreationDateTime="2014-11-
11T13:58:18+00:00" FileType="Snapshot" ODMVersion="1.3"
xmlns="http://www.cdisc.org/ns/odm/v1.3"     xmlns:OpenClinica="http://www.openclinica.org/ns/odm_ext_v130/v3.1"
xmlns:OpenClinicaRules="http://www.openclinica.org/ns/rules/v3.1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"      xsi:schemaLocation="http://www.cdisc.org/ns/odm/v1.3
OpenClinica-ODM1-3-0-OC2-0.xsd">
  <Study OID="S_VPH">
  <AdminData StudyOID="S_VPH">
  <ClinicalData StudyOID="S_VPH" MetaDataVersionOID="null">
        <SubjectData             SubjectKey="SS_STHJESSS"            OpenClinica:StudySubjectID="STHJessSmith"
OpenClinica:Status="available" OpenClinica:Sex="m" OpenClinica:EnrollmentDate="2014-09-04">
            <StudyEventData    StudyEventOID="SE_INITIALVISIT"    OpenClinica:StartDate="2014-10-13    00:00:00.0"
OpenClinica:Status="data entry started" StudyEventRepeatKey="1">
            <StudyEventData    StudyEventOID="SE_FOLLOWUPVISIT"    OpenClinica:StartDate="2014-10-13    02:09:00.0"
OpenClinica:Status="data entry started" StudyEventRepeatKey="1">
            <StudyEventData    StudyEventOID="SE_FOLLOWUPVISIT"    OpenClinica:StartDate="2014-10-13    00:00:00.0"
OpenClinica:Status="data entry started" StudyEventRepeatKey="2">
            <StudyEventData    StudyEventOID="SE_IMAGINGVISIT"    OpenClinica:StartDate="2014-09-04    00:00:00.0"
OpenClinica:Status="data entry started" StudyEventRepeatKey="1">
                <FormData FormOID="F_TABLEOFMODAL_V10" OpenClinica:Version="v1.0" OpenClinica:Status="invalid">
                    <ItemGroupData          ItemGroupOID="IG_TABLE_MODALITYTABLE"          ItemGroupRepeatKey="1"
TransactionType="Insert">
                        <ItemData ItemOID="I_TABLE_MODAL_TYPE_TABLE" Value="1" />
                        <ItemData ItemOID="I_TABLE_MODAL_DATE_TABLE" Value="2009-05-05" />
                    </ItemGroupData>
                    <ItemGroupData          ItemGroupOID="IG_TABLE_MODALITYTABLE"          ItemGroupRepeatKey="2"
TransactionType="Insert">
                        <ItemData ItemOID="I_TABLE_MODAL_TYPE_TABLE" Value="2" />
                        <ItemData ItemOID="I_TABLE_MODAL_DATE_TABLE" Value="2014-02-04" />
                    </ItemGroupData>
                </FormData>
                <FormData FormOID="F_ADDSUBJECTTO_V10" OpenClinica:Version="v1.0" OpenClinica:Status="invalid">
                    <ItemGroupData           ItemGroupOID="IG_ADDSU_UNGROUPED"           ItemGroupRepeatKey="1"
TransactionType="Insert">
                        <ItemData ItemOID="I_ADDSU_IMS_CONF" Value="1" />
                        <ItemData ItemOID="I_ADDSU_IMS_INIT" Value="kt" />
                        <ItemData ItemOID="I_ADDSU_IMS_DATE" Value="2014-09-02" />
                        <ItemData ItemOID="I_ADDSU_IMS_PDF" Value="" />
                    </ItemGroupData>
                </FormData>
            </StudyEventData>
        </SubjectData>
        <SubjectData SubjectKey="SS_STHTEST" OpenClinica:StudySubjectID="STHTest" OpenClinica:Status="available"
OpenClinica:Sex="m" OpenClinica:EnrollmentDate="2014-09-09">
            <StudyEventData    StudyEventOID="SE_INITIALVISIT"    OpenClinica:StartDate="2014-10-13    00:00:00.0"
OpenClinica:Status="data entry started" StudyEventRepeatKey="1">
            <StudyEventData    StudyEventOID="SE_FOLLOWUPVISIT"    OpenClinica:StartDate="2014-10-13    02:09:00.0"
OpenClinica:Status="data entry started" StudyEventRepeatKey="1">
            <StudyEventData    StudyEventOID="SE_FOLLOWUPVISIT"    OpenClinica:StartDate="2014-10-13    00:00:00.0"
OpenClinica:Status="data entry started" StudyEventRepeatKey="2">
            <StudyEventData    StudyEventOID="SE_IMAGINGVISIT"    OpenClinica:StartDate="2014-09-04    00:00:00.0"
OpenClinica:Status="data entry started" StudyEventRepeatKey="1">
        </SubjectData>
        <SubjectData  SubjectKey="SS_WOOD1"  OpenClinica:StudySubjectID="wood1"  OpenClinica:Status="available"
OpenClinica:Sex="m" OpenClinica:EnrollmentDate="2014-10-20">
            <StudyEventData    StudyEventOID="SE_INITIALVISIT"    OpenClinica:StartDate="2014-10-13    00:00:00.0"
OpenClinica:Status="data entry started" StudyEventRepeatKey="1">
            <StudyEventData    StudyEventOID="SE_FOLLOWUPVISIT"    OpenClinica:StartDate="2014-10-13    02:09:00.0"
OpenClinica:Status="data entry started" StudyEventRepeatKey="1">
            <StudyEventData    StudyEventOID="SE_FOLLOWUPVISIT"    OpenClinica:StartDate="2014-10-13    00:00:00.0"
OpenClinica:Status="data entry started" StudyEventRepeatKey="2">
            <StudyEventData    StudyEventOID="SE_IMAGINGVISIT"    OpenClinica:StartDate="2014-09-04    00:00:00.0"
OpenClinica:Status="data entry started" StudyEventRepeatKey="1">
                    </SubjectData>
        <SubjectData           SubjectKey="SS_STHTESTB_7448"           OpenClinica:StudySubjectID="STHTestBot"
OpenClinica:Status="available" OpenClinica:Sex="m" OpenClinica:EnrollmentDate="2014-11-04">
            <StudyEventData    StudyEventOID="SE_INITIALVISIT"    OpenClinica:StartDate="2014-10-13    00:00:00.0"
OpenClinica:Status="data entry started" StudyEventRepeatKey="1">
            <StudyEventData    StudyEventOID="SE_FOLLOWUPVISIT"    OpenClinica:StartDate="2014-10-13    02:09:00.0"
OpenClinica:Status="data entry started" StudyEventRepeatKey="1">
            <StudyEventData    StudyEventOID="SE_FOLLOWUPVISIT"    OpenClinica:StartDate="2014-10-13    00:00:00.0"
OpenClinica:Status="data entry started" StudyEventRepeatKey="2">
            <StudyEventData    StudyEventOID="SE_IMAGINGVISIT"    OpenClinica:StartDate="2014-09-04    00:00:00.0"
OpenClinica:Status="data entry started" StudyEventRepeatKey="1">
        </SubjectData>
    </ClinicalData>
```