OpenMRS Metadata Sharing Server: Connecting Academia and HFOSS

Joseph Wilder, 2012

Submitted to the
Department of Computer Science,
Princeton University,
towards fulfillment of the requirements
of Undergraduate Independent Work

Final Report

May 2, 2012

Advisor: Dr. Robert Dondero
Reader: Professor Brian Kernighan

This paper represents my own work in accordance with University regulations.
Acknowledgements

This has been an up and down year. I would never have gotten this far without the patience, insight, frustration, persistence, disappointment, and resourcefulness of my advisor Dr. Robert Dondero and my mentor from the OpenMRS project Rafal Korytkowski. Thank you to Darius Jazayeri for the initial inspiration of working on HFOSS and introduction to OpenMRS. Thanks also to Piotr Bryk for his work developing the OpenMRS Metadata Sharing Server. Special thanks go to my friends Nathan Keyes ’12 for hardware and moral support, to Erin Mills ’13 for assisting with formatting of this report and believing in me even when I didn’t, and to my parents, whose support from afar was greatly appreciated.
## Contents

1 Background .................................................................................. 1
   1.1 OpenMRS and HFOSS ................................................................. 1
   1.2 Metadata Sharing Module .......................................................... 4

2 Methodology ................................................................................. 7
   2.1 Getting to know an OSS Community and Distributed Development .... 7
   2.2 Software Development Tools ...................................................... 8
   2.3 Agile and Test Driven Development Software Processes .................. 10

3 Design ......................................................................................... 12
   3.1 Use Cases .............................................................................. 12
   3.2 Scenarios ............................................................................ 13
      3.2.1 Anonymous Users ............................................................. 13
      3.2.2 Authenticated Regular Users .......................................... 14
      3.2.3 Authenticated Administrator Users ................................... 16
   3.3 Software Requirements Specification (SRS) .................................. 17
   3.4 Initial Approach .................................................................. 19
   3.5 New Approach .................................................................... 22

4 Functionality ................................................................................. 24
   4.1 Needs of Users .................................................................. 24
   4.2 Implementation Walkthrough ................................................... 24
   4.3 Testing ............................................................................... 29

5 Evaluation .................................................................................... 30

6 Conclusion .................................................................................... 32
   6.1 Future Work ..................................................................... 32
   6.2 Conclusion ....................................................................... 33

A Global OpenMRS Distribution .................................................... 36

B Initial Project Schedule ................................................................. 40

C Example Metadata Package .......................................................... 41
   C.1 header.xml ....................................................................... 41
   C.2 metadata.xml ..................................................................... 43

D IEEE SRS Standard .................................................................... 47

E Database Schema ................................................................. 48
F  Code  50
F.1 Setup .............................................. 50
F.2 templates.php ................................. 51
F.3 utils.php ........................................ 56
F.4 Functions ........................................ 58
F.5 home.php ......................................... 59
F.6 newaccount.php ............................... 61
F.7 loginPage.php .................................. 62
F.8 profilepage.php ............................... 63
F.9 upload.php ....................................... 64
F.10 managePackages.php ..................... 65
F.11 manageUsers.php ............................ 67
F.12 search.php ..................................... 70
F.13 download.php .................................. 71
F.14 addUser.php ................................. 73
F.15 login.php ........................................ 75
F.16 uploader.php ................................. 76
F.17 peditor.php .................................... 79
F.18 ueditor.php .................................. 81
F.19 deletePackage.php ....................... 84
F.20 removeUser.php ............................. 86
F.21 Exit pages ...................................... 87
F.22 proc.php ........................................ 88
F.23 Javascript ...................................... 92
List of Figures

3.4.1 GUI mockup for uploading a package to MDSS. ......................... 20
3.4.2 GUI mockup for browsing and downloading packages from MDSS. .... 21

4.2.1 Screenshot of the OpenMRS MDSS homepage. ........................ 26
4.2.2 Screenshot of the OpenMRS MDSS user registration page. ............ 27
4.2.3 Screenshot of the OpenMRS MDSS profile page. ........................ 27
4.2.4 Screenshot of the OpenMRS MDSS upload page. ........................ 28
4.2.5 Screenshot of the OpenMRS MDSS homepage. .......................... 28

A.0.1 A map of current OpenMRS clinical and research sites ............... 36
A.0.2 A table listing the current OpenMRS clinical and research sites .... 39

B.0.1 The initial project management schedule. .............................. 40
Abstract

Implementation of software to gain hands-on experience with OSS communities and software engineering processes in support of the OpenMRS project. Deployments of the open source medical records software application OpenMRS can create and export/import metadata but do not have a good centralized way to share it. The Metadata Sharing Server (MDSS) was conceived as an answer to that obscurity and inefficient duplication of effort. MDSS can be thought of as an OpenMRS specific type of package manager way for community members from over 26 countries to pool their metadata resources in support of advancing medical care in situations where enterprise level software solutions are infeasible or undesirable. Major lessons involved getting accustomed to distributed collaboration, navigating the multitude of tools in use in OSS projects, and introduction to Agile and test driven design methodologies. A thesis project for academic requirements and ‘...In the service of all nations’.
Chapter 1

Background

OpenMRS, as the acronym suggests, is an Open Source Medical Records Software application. The project technically falls into a subset of open source software projects known as humanitarian free open source software (HFOSS, http://hfoss.org). HFOSS and software engineering is not currently a part of the Princeton University undergraduate computer science curriculum. Consequently, working with OpenMRS involved not just design and implementation of software but also a large and novel component of exposure to and learning about open source software communities, distributed software development and collaboration, an alphabet soup of OSS tools, and Agile software engineering processes. The core idea and emphasis which led to seeking out a project in HFOSS to complete independent work is to seek out dynamic work that won’t end up gathering dust on a shelf. Instead, the ideal is a real world project to gain experience, which positively impacts the lives of real people, and bridges between academia in the university and software engineering in the wild. A project that would truly live up to the broad scope of Princeton’s motto ‘Princeton...in the service of all nations.’

1.1 OpenMRS and HFOSS

The OpenMRS project is a multi-institution non-profit collaborative software and humanitarian project open to all who are interested in contributing, including many organizations such as international and government aid groups, NGO’s, and for-profit and non-profit corporations, as well as the many individual contributors [1]. The project is led by led by the Regenstrief Institute, a world-renowned leader in medical informatics research, and Partners In Health, a Boston-based philanthropic organization with a focus on improving the lives of
underprivileged people worldwide through health care service and advocacy. The mission of OpenMRS is to improve health care delivery in resource-constrained environments by coordinating a global community that creates a robust, scalable, user-driven, open source medical record system platform. OpenMRS is a software platform and application which enables design of a customized medical records system. No programming knowledge is necessary to use OpenMRS (although medical and systems domain knowledge is required). OpenMRS is a common platform upon which medical informatics efforts in developing countries can be built. The OpenMRS software is free to download and released under the GNU Public License (GPL). HFOSS projects like OpenMRS include elements of both free software (FS) and open source software (OSS). Generally, discussions about the application of these terms have become almost a fundamental philosophical discourse and reach far beyond just developing software.

Implicitly, it reflects the choice between two different fundamental self-perceptions, aligned with different life-styles and political conceptions of the world.

According to members of the FS community, the intended meaning of "Free Software" is "software that gives the user certain freedoms", but the term also invites to the unintended interpretation as "software you can get for zero price" (cf. www.gnu.org/philosophy/free-software-for-freedom.html). These freedoms contain ethical issues, aspects of responsibilities and of convenience. [2]

On the other hand,

Members of the OSS community define "Open Source Software" as software that allows everybody to have a look at its source code and stress the practical benefits of such software, while aspects of freedom are rather neglected in the definition. "Open Source Software" contains a broader variety of software than it is allowed by the term "Free Software", it comprises free software as well as semi-free software and even certain proprietary programs. [2]
OpenMRS, then, is truly HFOSS because its source code is open and has practical benefits, primarily that it is available for zero cost and it gives developing or underprivileged people more freedoms in access to modern healthcare. Since being created in 2004 OpenMRS has spread far and wide through the work of its collaborative community. See Appendix A for more information on OpenMRS research and clinical implementation sites in more than 26 countries worldwide.

OpenMRS is written in Java and HTML, and is designed to be cross platform. The application handles a variety of use cases from the level of national health infrastructure to a local field clinic. To be adaptable to any context that might arise, individual customized setups of OpenMRS for specific clinical contexts are created. These unique setups are known as implementations, and are identified by their implementation ID number. Members of the OpenMRS community who specialize in creating new implementations are referred to as implementers. These and other community groups are further discussed in Section 2.1. The platform itself is a data model surrounded by an abstracting API layer and then paired with a web-based app utilizing a client-server design to provide access to the electronic medical records. The OpenMRS data model is in practice treated as a black box, but is based around the idea of a central ‘concept dictionary’ which handles operations on information codified as ‘concepts’ independent of the type or any particulars of the information collected and operated on. This model is descended from the Regenstrief Institute EMRS model first developed by the eponymous Indianapolis-based international biomedical informatics and healthcare research organization in 1994 [1]. Unique implementation configurations are enabled not just by being able to handle operations on diverse concepts and running in the JVM over multiple hardwares, but also by the ability to include a variety of modules and metadata. Modules are community developed add-ons to the application containing additional libraries and functionality for particular tasks or situations. The word metadata refers a whole class of data including members in multiple categories such as: a) concepts, b) locations, c) encounter types, d) user roles, e) programs, f) new drug regimens, g) treatment
plans, h) language packs, or i) clinical encounter forms. Metadata do not include individual electronic patient records. Some metadata are pre-installed as part of configuring an implementation, but more often metadata are created by users of an implementation in the course of their work. Metadata are also created and improved on by users of the OpenMRS development community at large. To contribute, collaborate, and avoid redundancy those users then often want to share their metadata with other members of the community who might be tackling a similar challenge or situation. Currently the best mechanism that exists for sharing metadata is the Metadata Sharing Module.

1.2 Metadata Sharing Module

The Metadata Sharing (MDS) Module is an add-on module to OpenMRS that facilitates exporting and importing of metadata. It was initially implemented in two phases as Google Summer of Code projects in 2010 and 2011 and is currently in beta version 0.10.x. The module, when installed, is accessed through the OpenMRS application’s Administration panel. Users can choose to create a new metadata package containing a selection of any of the locally installed metadata and their dependencies. Users can also manage any of their previously created packages or import a new package sourced from some other channel. This system is effective at these tasks but has limited spread and networking as a byproduct of its individual and distributed nature which often leads to frustration or duplication of efforts when users are not aware of or do not have access to the previous work of other metadata developers, implementers, and users. The next paragraph will go over more detailed usage of the MDS module operations.

When choosing to create a new metadata package with the MDS module, the user is prompted to select which metadata from all available metadata in the current implementation should be added to the package. Users then give the package a name and description and the module packs the chosen metadata, as well as any concept or other metadata dependencies of the chosen metadata, into an XML file called metadata.xml. The module
then creates a header file for the package containing the package name, package version, package description, package uuid, creation date, OpenMRS version, and group. See appendix E for example metadata package XML files. These two XML files together are then bundled into a zip archive which is the metadata package. All exported packages are then published and available for download through the web-based app as a REST service from a URL of this form: http://%yourhost%:8080/openmrs/ws/rest/metadatasharing/package/%UUIDOFPACKAGE%/%VERSION/download. Managing previously exported packages simply entails unpublishing or republishing those packages to their URLs. Importing a package requires having either the zip archive of a metadata package or having an active published URL of the package that you would like to download. When importing it is possible to choose different levels of trust which configure whether or not local metadata and concepts will be overwritten with imported metadata and concepts. A warning will be displayed if a user tries to import a package exported with a different version of OpenMRS or if a newer version of the package being imported is already installed. A list of all metadata in the package is then shown and options are given for each piece including: a) create new, b) skip if possible, c) keep mine, and d) overwrite. After that the metadata is imported. Note that uuids and conceptids stay constant, if overwriting occurs only the content is overwritten. Managing already imported packages gives an option to check for a new update of a package from a published URL or to ‘subscribe’ to a package and set a timer for how often the MDS module will automatically check the URL for an updated version of that package. The obvious limitations of this functionality lie in accessing metadata to import and include the necessity of either somehow having a package to start with or having a working internet connection and knowing the URL of any possible exporters of the package who are currently actively publishing it.

The OpenMRS MDSS then is envisioned as an assistant to the MDS module consisting of a web application acting as a central repository of metadata packages exported from different users’ MDS modules. The server will centralize, publicize, and standardize packages
of metadata so that all OpenMRS implementers, developers, and users can easily access or peruse all available metadata packages created by the community globally without having to find out that a package exists, track down someone who has the most recent version of a package, and then receive a copy of the package or the published URL of the package from the exporter in a different channel. This will broaden awareness of the available metadata, reduce metadata creation redundancy, better communicate community metadata needs, and allow easier tracking of multiple versions of a given package. In addition, exporters will no longer need to continuously expose their base application to publish packages and may use less of their internet bandwidth for a wider level of distribution.
Chapter 2

Methodology

2.1 Getting to know an OSS Community and Distributed Development

A large part of this project is getting real world experience and learning about how OSS projects and communities operate. In the OpenMRS organization, community members who work on development are divided into three main groups. The first is a small group of core developers who are intimately familiar with the main project, devote much of their time to it, and are continuously involved in discussing, designing, and implementing new changes and updates for it. This group is also the primary source of decisions about the direction of OpenMRS and setting the road map and milestones, although those decisions are also made in consultation with the community at large and other important players. A larger group also exists of regular developers who are less involved in the project but still active in the code base. Many times regular developers may be professionals volunteering their spare time or students trying to fulfill university requirements. Regular developers often work only for a set season or shorter period of time or may focus only on one ‘pet project’, feature set, module, or aspect of the software. In some organizations core developers or regular developers can be paid for their work even if the software they work on is free, but this is not the case with OpenMRS. A third group of developers are casual developers. These developers often take on introductory tasks or seek heavily the mentorship of more experienced developers. Casual developers tend to be relatively short lived, either advancing to be a regular developer or transiently moving on to other projects. Some of the more long lived casual developers are active members of other parts of the OpenMRS community but simply don’t have a
programming background or interest in software development beyond what they need to use OpenMRS. Another important user group within the OpenMRS community are the implementers. These advanced users take the application and set it up, adapt it to particular local systems and needs, train other users (especially non-technical users), and maintain it in various clinical and hospital settings around the world. The implementer and developer groups each have their own email mailing lists and weekly conference calls, but they are not mutually exclusive and many of the best development suggestions come from users who are in both groups.

OSS development projects that are distributed especially internationally like OpenMRS are referred to as DOSSD projects. Because the OpenMRS project is open to all who are interested in contributing, including many organizations such as international and government aid groups, NGO’s, and for-profit and non-profit corporations, as well as the many individual contributors, the interested parties are highly scattered geographically and internationally over the whole world. OpenMRS is then a prime example of an DOSSD project and a distributed OSSD community. When a collaborative community is so dispersed many different methods of communication must be utilized to enable software development and collaboration.

### 2.2 Software Development Tools

Tools which are themselves OSS are preferred for enabling collaboration on OSS projects like OpenMRS and many software companies will give special licenses to OSS projects and their developers. Companies which have given OpenMRS special OSS licenses to their tools include Atlassian, AquaFold, Balsamiq Studios LLC, Bitrock, Blueberry Software, JetBrains, Inc., Napkee Labs, Yourkit, and Zero Turnaround. Many of the tools used (both licensed and OSS) are new to me coming from the experience of largely individualistic academic classwork but their purposes are common knowledge to anyone who has worked in software development, especially distributed software shops.
The primary tool that aids distributed development is source code control. OpenMRS uses Subversion to keep track of their repository and code edits, patches, branches, and permissions. They suggest that users interact with Subversion through various pre-tested IDE plug-ins, graphical interfaces, and command line tools such as Subclipse and Tortoise. Use of other clients for interacting with the Subversion repository is up to the discretion of the individual developer but may not be supported. At the same time as this project was taking place investigations were being conducted by the community to begin the process of transitioning to use Git for source control in place of Subversion. The first code branches to be moved to Git were the main application and certain key ‘core’ modules adopted for maintenance by the core developers. Git is preferred to Subversion, as long as the transition is manageable, because it simplifies permissions and access control. In Subversion, permissions are time consuming because each and every developer must be added to the user class which has permissions to interact with the particular branch they want to work on. In Git, only a few very active core developers need permission to push code changes to trunk of core and everyone else can easily fork a branch with no permissions necessary and then submit a pull request to the core developers when they are done working. This setup provides a lower barrier to entry for new developers to try out their ideas quickly. Git is also preferred because as a distributed repository it has better support for ‘commit’ operations in conditions with unstable or intermittent internet connections, as is typical in developing countries. Subversion is preferred because it is much simpler to learn at first and has the momentum of already being established in use. It is likely that OpenMRS will soon follow the growing trend of OSS projects moving to Git, although the effort to make the switch, update all the documentation, and educate the users will not be insignificant.

After source control in the tool importance hierarchy comes the primary issue tracking tool JIRA. JIRA is useful for keeping track of all phases of software development process through entering, assigning, and taking actions on tickets that track through the necessary phases as well as providing a platform for discussions and code reviews. JIRA ties in with
Subversion and Git so that committed code changes are hotlinked to the tickets they pertain to. Additional information is also generally referenced from articles on the OpenMRS wiki. Dashboards can then be created which track progress on a given class of tickets, on tickets related to a given section of code, or on tickets labeled belonging to a given sprint.

The final most important type of tool in DOSSD is a whole class of communication technologies for collaboration. When developers are in different timezones around the world the number one communication tool is email but OpenMRS also successfully utilizes their documentation and discussion wiki, their own IRC channel, Skype, and other video and teleconferencing applications such as Youtube, Adobe Connect, Breeze, and Etherpad. These tools allow for a variety of meetings and more or less in-person, mostly synchronous communications. Daily scrum meetings from the core developers involved in any active sprints are a highly visible example of these technologies in practice. OpenMRS University sessions aim to teach skills or answer questions of new users and developers interested in getting involved. Weekly design calls and implementers calls serve as forums of discussion for their respective groups for decisions that are better handled in person instead of over endless email threads.

2.3 Agile and Test Driven Development Software Processes

OpenMRS favors a modern style of software engineering process which is similar to Agile Programming and Test Driven Development. This class of processes are often called extreme programming and rely on rapid iteration and flexibility in the process. The key principles of this approach include: a) stakeholder involvement, b) incremental delivery of intermediate products, c) expectation of change, d) fluid requirements, and e) active emphasis on promoting simplicity and reducing coupling ([3] chapter 17.1). Each developer is free to follow their own individual preferred process, but the organization as a whole is very flexibility-focused. The sprint process OpenMRS uses, by which all effort is concentrated on a set of well-defined goals for a short period of time, is very much a hallmark of these kinds of processes. Daily
scrum meetings by the core developers which emphasize quick reporting of what has been done in the last day, what work is being done now, and what project blockers if any are being waited on, is another example of extreme programming in OpenMRS. The OpenMRS application is built such that any changed code when rebuilt is run against extensive Unit Tests through Jrebel. These tests were written before the classes that they test and help make sure that unexpected breakages and/or work stoppages due to committed code with bugs are as rare as possible. OpenMRS, also as part of their Agile-like process, (at least for the core developers) ascribes to a philosophy of being as fluid in design as possible, as seen in their use of many mock-ups and rapid prototypes. These iterative and less-defined processes are quite an experience and very different from the code development experience available in my undergraduate curriculum.
Chapter 3

Design

Beyond learning about the OSS community, its many tools, and software engineering and Agile methodologies, the primary goal of the project was to develop software for the MDSS. The first step in building the software was to define the goal. The definition process for this project started with text-based use-cases (as opposed to use-case diagrams in UML), which were further explored in individual scenarios, which in turn were specified by eliciting criteria. This was followed by construction of requirements, and finally the proper tools were selected. This process was careful to take into account and accurately represent the goals, domain knowledge, and stakeholders of the MDSS software ([4] chapter 2).

3.1 Use Cases

Primary use cases were the first step in defining the MDSS. There are three main actor classes in these use cases: 1) an anonymous user, 2) an authenticated regular user, and 3) an authenticated administrator user. These user classes are listed here hierarchically in that each class shares all the use cases of the classes that are less access restricted. Anonymous users have three main uses of the MDSS: browsing available packages, downloading a package, and registering as a new user. They can also transcend actor classes by logging into an existing account. Regular users then share all of those use cases. There are four additional uses available to authenticated regular users: 1) uploading a package, 2) interacting with their own uploaded packages, 3) editing their own user account, and 4) logging out. Authenticated administrator users then share all of the use cases of anonymous and authenticated regular users and, in addition, have two use cases of their own. Only authenticated administrator users can interact with all users’ uploaded packages and accounts. Each of these use cases
encapsulates a set of different actions, so after describing the MDSS use cases I proceeded to tease out individual interaction threads ([3] chapter 7).

Next, scenarios were constructed to describe the individual actions possible in each use case. These scenarios, given below grouped by the use case to which they belong, capture another, deeper level of details about the system but are still relatively informal in order to be easy to relate to and explore. These scenarios lay out the key capabilities of the software and the desired patterns of interaction with the three different actor classes, as well as what can go wrong in each of the situations and how that might be handled ([4] Knowledge Area 2).

3.2 Scenarios

3.2.1 Anonymous Users

Initial assumption: User with a properly functioning OpenMRS implementation, MDS module, an internet connection and browser, and the address of a MDSS.

Browsing available packages A user navigates to a page with a list of available uploaded packages. The user can view multiple types of data about each package, sort the list, and search through the list.

Things that can go wrong: If search returns no results the user is notified and continues browsing.

Downloading a package When looking at a list of packages a user can click on a download link and be prompted for a path to save the selected package, which then downloads.

Things that can go wrong: If a user tries to open a download link to an invalid or no longer available package they are notified that their download link was invalid and there is no such package, then the user is returned to browsing. If the selected location is invalid or the connection is interrupted the download will fail but the MDSS is not responsible for handling these failures
Registering as a new user account A user navigates to a registration form and enters a username, email address, and password. The user is registered, notified of successful registration, and then prompted to try logging in.

Things that can go wrong:: If a user doesn’t enter any of the three pieces of required info they are prompted to enter the required info and returned to the form. If a user enters a username or email address already registered to another user they are notified that those credentials are already in use and returned to the form.

Logging into an existing account A user navigates to a login form and enters credentials of an existing account, is logged in, welcomed, and shown the account details of their account.

Things that can go wrong:: If the account entered does not exist or the password entered is incorrect the user is notified generically that they typed something wrong and returned to the login form.

3.2.2 Authenticated Regular Users

Uploading a package An authenticated user can navigate to an upload form where they are prompted to choose a file to upload. The package is then uploaded and the user provides details about that package for display.

Things that can go wrong:: If the file does not have the correct characteristics the file will fail to upload. The user will be notified and returned to the upload form. If no details are available the upload will fail, the user will be notified and returned to the upload form.

Interacting with own uploaded packages An authenticated user can see a list of all their own uploaded packages. They will be able to navigate to a form to edit the entered details any of those packages. The details of a package that are editable will
be displayed and when finished editing a user will confirm saving any changes. They will also be able to remove those packages from the server.

*Things that can go wrong:* If a user has not successfully uploaded any packages the list will notified to be empty. If a user begins editing the details of a package but does not finish or confirm the edits they should not be saved. A user may edit the unique details of a package to be the same as another package already shared or a user may edit a field (such as subscription url) to a value that is not valid for that field type. In those cases the user should be notified that they have made an illegal/restricted edit and be returned to their uploaded packages list.

**Editing with their own account** An authenticated user can navigate to an editing form for the details of their user account. The current details which are editable should be displayed. The user can then edit them and when finished confirm and submit the changes and then be redirected to their user page.

*Things that can go wrong:* A user may begin editing their details and stop in the middle, in that case the edits should not be saved. A user may edit their unique details to be the same as another registered user or a user may edit a field (such as email) to a value that is not valid for that field type. In those cases the user should be notified that they have made an illegal/restricted edit and be returned to their user page.

**Logging out** An authenticated user can, from any page on the site, choose a log out option which will terminate their session and return them to the home page as an anonymous user.

*Things that can go wrong:* If no user is found to be logged in when the logout option is activated it will silently redirect to the home page with no other changes.
3.2.3 Authenticated Administrator Users

Interact with all users’ uploaded packages An authenticated administrator user can navigate to a listing of all uploaded packages. They should be able to sort these packages by different criteria and search through these packages. For any package in that list, an authenticated administrator user will be able to navigate to a form for editing the details of that package or removing that package from the server in the same way that any authenticated user could for their own uploaded packages in Section ?? above.

Things that can go wrong: The same things that could go wrong for individual authenticated users interacting with their own uploaded packages above, except that the redirect will be to the administrator’s listing of all uploaded packages instead of to the authenticated user’s own profile page. An administrator can attempt to edit their own account details to remove their admin status. This operation is unusual but should succeed because at the time of the transaction they have permission to make that edit.

Interact with all users’ accounts An authenticated administrator user can navigate to a listing of all registered users. For any user listed they can navigate to a form for editing the details of that user’s account. They can also remove any user from the server.

Things that can go wrong: The same things that can go wrong with a user editing their own account details above except that the redirect should be to the administrator’s list of all users instead of to the user’s own account details page. An administrator tries to remove themselves as a user from the server. This is an unusual case but should succeed because at the time of the operation the administrator has permission to do so.
3.3 Software Requirements Specification (SRS)

The next step in defining the software is generating a software requirements specification (SRS) document from the use cases, scenarios, and more detailed criteria. The SRS is an official statement of what the developer (in this case, me) should implement. I wanted to define a more formal SRS than most Agile processes would use (they even go so far as to recommend having a set of cards which can be constantly reorganized by the customer according to their changing priorities) because I saw meeting my requirements contract as one of my primary criteria for success. A more standard formal SRS specification is IEEE/ANSI 830-1998 (See Appendix D). This specification on the other end of the spectrum has way too much detail for this type of project, but it served as an example of a formal document for requirements. The type of SRS is dependent on the type of system being developed and the development process being used. In my case I am part a very small team in a very iterative process on a small scale system. Therefore I can have a more flexible, less detailed SRS and count on resolving any ambiguities when I reach them in implementation ([3] chapter 6).

Server Generics

1. The server will run on any operating system.
2. The server will use PHP/HTML as its primary language.
3. The server may also use javascript as necessary.
4. The server may also use CSS as necessary.
5. The server will interact with a MySQL database.

Uploading

1. The server will accept uploaded metadata package files which are of MIME types application/zip, application/x-zip-compressed, multipart/x-zip, and application/s-compressed.
2. The server will accept uploaded metadata package files which are up to 1MB in size.
3. The server will not accept uploaded php files (these are dangerous security hazards).
4. The server will rename any uploaded files to a random name, for security reasons.
5. The server will be able to save uploaded files outside of the webserver root, for security reasons.
6. The server will check for name, OpenMRS version, and package version collisions before accepting an uploaded package.
7. The server will verify that the header.xml and metadata.xml files are properly-formed XML before accepting a new upload.
8. The server will index package headers for information including name, uuid, description, publication date, publication url, OpenMRS version, and package version.

Editing Packages
1. Users will be able to edit the name and description details of their own uploaded packages.
2. Administrator users will be able to edit the name and description details of other users’ uploaded packages.
3. Administrator users will be able to remove any uploaded package from the server.
4. The server will not allow package detail edits which cause a name, OpenMRS version, and package version collision with an already shared package.
5. The server will show a confirmation or failure message for attempted package detail edits.
6. The server will persist package detail edits, not only in the database, but also in the header.xml file itself.

Passwords
1. All passwords will be stored in the database, hashed for security.
2. All hashed passwords will be salted to guard against rainbow table attacks.
3. All password must be at least 8 characters in length.

Browsing shared packages
1. The MDSS will list packages available including information about name, description, author, date uploaded, package version, OpenMRS version, and download count.
2. The listed packages will be orderable by popularity (download count), recency of upload, author, OpenMRS version, and name.
3. The listed packages will be searchable by name, description, or author.
4. All displayed packages will have corresponding download links.

User accounts
1. New users will be able to register with the site.
2. Registration will collect username, email address, and password.
3. No two users will be able to register the same email address.
4. Users will be able to use their email and password credentials to authenticate themselves and log in.
5. The server will show an error message for failed authentications.
6. All pages will contain a link to log in.
7. There will be at least two user classes with different privileges: regular user and administrator user.
8. All pages visible to authenticated users will contain a link to log out.
9. Users will be able to edit their own account details, including name, email, and password.
10. The server will give an error message and not save an edit if it causes a name or email collision with another registered user.
11. Administrator users can edit the account details of other users, including name, email, password, and user class.
12. Administrator users can reset the password of any user to a default password.
13. Administrator users can remove any user from the server.
14. If a user is completely removed by an administrator, all of that user’s uploaded packages are also removed.

3.4 Initial Approach

One of the initial tasks for any software development project management plan is to plan out a schedule. This was very difficult to do on this project. I planned out an initial schedule which listed large milestones such as Learning Tasks, Version 1.0: Implement Server, and Feature Set 1. The slightly improved but still entirely naïve schedule is included in Appendix B for your amusement and chagrin. This projection was in terms of naïve feature wishes, but had no realistic concept of what it would take to reach those and in which order they might necessarily be developed. Beyond the initial idea and a really good story, I had difficulty anticipating what kinds of steps might be necessary in the process of developing my first server. The target project schedule was one that involved very granular tasks to the level of days and keeping track of projected days to completion of each step and actual days to completion to track project overrun. The initial idea was to develop in Java and this lead to an initial schedule with a large portion of time spent on learning tasks about Java and the other tools necessary for Java servers. I also spent a lot of time getting OpenMRS trunk checked out and set up for development in Eclipse and running locally on Jetty. My third major task was beginning to create GUI mock-ups for the MDSS in Balsamiq which
is integrated with JIRA. The OpenMRS principles guiding gui design include: simplicity, unclutteredness, self-explanation, and least number of clicks for the most common tasks. See Figures 3.4.1 and 3.4.2 for examples of some of the resulting mock-ups. Progress on the Java front was very slow on my own and repeatedly overshot estimates up until mid-December. At that point we tried implementing a new schedule where there was no fixed overall schedule, appropriate goals would be created based on progress and success was determined on a weekly basis of tickets or tasks, skype calls, and call summary notes. This system got some work going, but it was slow and difficult and often the weekly evaluations backfired into ignoring everything and prevented more attention from being paid to the tasks at hand especially during university breaks and exam periods early in the year.

Figure 3.4.1: GUI mockup for uploading a package to MDSS.
In the end the Java approach turned out to be too complicated with its multitude of tools (Eclipse, Maven, AppFuse, Spring, Jetty, Hibernate, etc.) that needed to be configured without getting familiar with their particulars and work along the way but were incidental to the project at hand. Eclipse should be familiar to any developer as one of the premier IDEs for working in Java. Maven is a dependency repository manager and building tool that is very helpful when the projects get as large as OpenMRS and depend on many different resources to all work together. Jetty is the lightweight java based HTTP client/server that is bundled in Maven. AppFuse is quickstart scaffolding generator which initiates a sample project and all of the structure in place in order to use Spring and Hibernate. Spring is a MVC framework for building web applications. Hibernate is an ORM library for Java so that persistence can be done regardless of the underlying database setup. The hardest part of using so many tools has been that the expectation is not to get to know each tool fully but instead to get each configured right so that they contribute to development of
the end goal with minimal intervention on the part of the developer. Each tool is not to be learned per se but rather picked up enough to quickly do what needs done always with an eye toward where the project needs to be going. I am not familiar with this approach coming out of the academic context where things are taught and learned thoroughly. This approach is necessitated by the scale of the project complexity, rapidly changing nature of Agile development, and the time frame of the proposed project such that everything should not be written from low level but instead recycled, and utilizing freely available tools is absolutely necessary.

3.5 New Approach

The low level of collaboration on my part in the project at this time made navigating these tools independently impossible and meant that a different approach was needed. Therefore, a third of the way through spring semester a new approach was hatched to use PHP for development which is more comfortable, very well documented, and requires much fewer supporting technologies although it doesn’t integrate as well with the OpenMRS platform itself. The new focus was on getting functionality first as quickly as possible and then getting minimally to the SRS however possible. The new approach needed new tools with a target on achieving functionality. For the requirement that the MDSS be fully portable the initial proposal to write the server in Java made a lot of sense because it could run above the hardware in the JVM. When I switched tracks to focus on PHP that meant finding a way to deploy my server cross platform. The solution that I found was to develop using XAMPP. XAMPP is a free cross platform Apache, MySQL, PHP, and Perl webservice stack. The server side code is written in PHP/HTML and interacts with a MySQL database (see Appendix E for database schema). The decision was made to do a majority of the operation on the server side for security but there will also be some client side operations. For achieving functionality on the client side the MDSS will use Javascript and a smattering of CSS. Some Javascript tasks such as form validation are simple enough to code directly, but for others such as
search functionality, Ajax requests for table pagination, and tabbed content presentation, a Javascript library will be necessary. The Javascript library for this project will be JQuery with some assistance from JQuery UI. There was also an attempt to separate out code for logic from code for presentation (see Appendix F for all PHP code). The MDSS is written first in English just like OpenMRS but later there may be opportunities to translate the interface. The database is collated in utf_8_general so that packages written in non-Roman alphabets can be supported for upload and hosting. Security is a large aspect of design of the MDSS. Because there are authenticated and non-authenticated users care was taken that all pages behave gracefully even if accessed directly by a non-authenticated user. Several security best practices are integrated in the design of the MDSS. All inputs that enter MySQL queries are escaped. Passwords are hashed, not stored in plain text, and those passwords are first salted against rainbow table attacks on a stolen database. File uploads are configured to be stored outside the webserver root, have a maximum file size, a mime type whitelist, and a file extension blacklist. XML files in packages are checked for proper form. In order to progress in more gradual steps even when I could not predict with fine enough granularity what tasks would need doing throughout the rest of the project I made renewed use of the JIRA ticket system. In this case Rafal, my mentor from OpenMRS, would make tickets for tasks he knew I would need to complete next and then I committed code to the repository linking each revision to a particular ticket. After several tickets were complete a code review would be started for all the code checked in during those tickets. The code review system in JIRA makes use of Crucible and FishEye to diff, comment, and flag changes. In all 16 tickets in JIRA were completed for this project although commits outnumber them 5 to 1. Other metrics used included: 1) three code reviews completed, 2) around 2000 lines of code written (growing and then shrinking as I factored out more common functionality; code is included in Appendix F), and 3) code quality measures which will be discussed in Section 4.3.
Chapter 4

Functionality

4.1 Needs of Users

The minimal needs of the users include the following:

1. Users need to be able to upload packages via the website with the ability to enter text fields such as name, description, and date published, which will describe the package. Users need to authenticate themselves in order to do this.

2. Users need to be able to (without authenticating themselves) browse a display of available uploaded packages including information about the packages’ a) name, b) description, c) authority, d) date published, e) download count, and f) download link.

3. Users need to be able to filter browsed packages by id, name, description, and authority with a single search field.

4. Users need to be able to see the top 10 downloaded packages without logging in.

5. Users need to be able to register as a new user, edit their user details (name, e-mail, password), and display/edit/remove their uploaded packages.

6. Some users need the ability to log in as an administrator and do the following: edit package details, remove any or all packages, remove users, and reset passwords.

4.2 Implementation Walkthrough

Broadly, the system works based on PHP scripts (see Appendix F) which interact with a MySQL database (see Appendix E) and generate HTML and Javascript pages which take input from users and hand it off to other php scripts to operate on.

There are five main and two restricted display pages in the MDSS. They all start with the MDSS header and end with the MDSS footer. The home page has navigation links, uploaded
packages search box and results area, and tabbed displays of tables of uploaded packages. The register page has inputs for submitting a name, email, and password to register and will indicate that they are required if any of them are submitted empty. The login page has space to input an already-registered email and password combination followed by navigation links. The upload page has navigation links and a file selection dialog with an upload button. The profile page displays navigation links, account details and an edit link, and tabs displaying that user’s uploaded packages. There are also the ‘edit user’ details and ‘edit package’ details dialogs which are subsets of the administrators-only ‘Manage Users’ and ‘Manage Packages’ pages, which have navigation links and list all records of their respective types with editing links. The site was implemented to match the requirements, because those requirements were constructed from the minimum user needs.

Without authenticating, users can browse the available packages on the home page in two ways: by searching for a name, description, or author through the search box, or by selecting one of the preloaded tabs for organized results. The initial tabs are: Newest (top 10 by upload date, not publishing date), Most Popular (top 10 by number of downloads), Alphabetical (all packages organized by name), By Author (all packages organized by author), and By Release (all packages by OpenMRS version). Each of these areas also shows columns for other info including a) name, b) description, c) authority, d) date published, e) package version, f) OpenMRS version, g) download counter, and h) download link. (See Figure 4.2.1). New users can register by giving a name, email, and password. Users can register a new account so long as no other user has previously registered the same name or user the same email address. (See Figure 4.2.2). Users can login to authenticate. When logging in, users are redirected to their profile page. From a user’s profile page, they can a) see and edit their account details including name, email, and password, b) see any packages they have uploaded, c) download packages, d) edit name and descriptions of packages (which are persisted in the header.xml file), or e) remove those packages. (See Figure 4.2.3). When logged in, users can upload new packages. The header.xml file is parsed for the package’s
data, including name, description, publication date, package version, and OpenMRS version. (See Figure 4.2.4). Users who are in the admin user class will also have additional navigation links to ‘Manage Packages’ and ‘Manage Users’. In ‘Manage Packages’ admin users can see all uploaded packages and download, edit name and description, or remove them. These changes are persisted in the header.xml file. In ‘Manage Users’ admin users can a) see all registered users, b) reset their password to a default of ‘MDSSUser-name’, c) edit their name, email, password, or class directly, or d) remove them. (See Figure 4.2.5). If an admin removes a user, all their uploaded packages are also removed.

Logged in users should log out when finished.

![OpenMRS](image)

**Metadata Sharing Server**

**Home**

---

**Uploaded Metadata Packages:**

<table>
<thead>
<tr>
<th>name</th>
<th>description</th>
<th>authority</th>
<th>date published</th>
<th>version</th>
<th>OpenMRS version</th>
<th>downloads</th>
<th>actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Really Long Name of the Coolest Metadata Package Ever!</td>
<td>Test package 3 of 4.</td>
<td>a</td>
<td>2011-12-02</td>
<td>1</td>
<td>1.9.0-SNAPSHOT</td>
<td>3</td>
<td>DL</td>
</tr>
<tr>
<td>Drug Resistant TB Plans</td>
<td>Test package 4 of 4.</td>
<td>test</td>
<td>2011-12-02</td>
<td>1</td>
<td>1.9.0-SNAPSHOT</td>
<td>2</td>
<td>DL</td>
</tr>
<tr>
<td>HTML</td>
<td>Test package 1 of 4.</td>
<td>admin</td>
<td>2011-12-02</td>
<td>1</td>
<td>1.9.0-SNAPSHOT</td>
<td>1</td>
<td>DL</td>
</tr>
<tr>
<td>Espanol</td>
<td>Test package 2 of 4.</td>
<td>Bob</td>
<td>2011-12-02</td>
<td>1</td>
<td>1.9.0-SNAPSHOT</td>
<td>1</td>
<td>DL</td>
</tr>
</tbody>
</table>

Figure 4.2.1: Screenshot of the OpenMRS MDSS homepage.
Metadata Sharing Server

New User Registration

Please fill out the following fields:

Name: test
Email: test@MDSS.org
Password: ******
Submit

Already registered? Login [here]
Home

Thanks for using Metadata Sharing Server!


Figure 4.2.2: Screenshot of the OpenMRS MDSS user registration page.

Metadata Sharing Server

Welcome, test!
You are in the admin user class!

Manage Packages
Manage Users
Upload
Home
Login

Name: test
Email Address: test@first.com
Member Since: 2012-04-30
Packages Uploaded: 0
Edit

Your Uploaded Packages:

Figure 4.2.3: Screenshot of the OpenMRS MDSS profile page.
Metadata Sharing Server

Welcome, test!

Upload a Package to Share:

Choose a file:

![Upload Package]

You are in the admin user class!

Manage Packages
Manage Users
Upload
Home
Profile
Logout
Login to a different account
Register a new account

Thanks for using Metadata Sharing Server!

Figure 4.2.4: Screenshot of the OpenMRS MDSS upload page.

Metadata Sharing Server

Welcome, test!

You are in the admin user class!

Home
Your Profile
Manage Packages
Logout

Editing User 3:

Name: j
Email: j@p.com

New Password
Confirm New Password
User Class:
C Admin
@ User
Save Changes
Cancel

(note: changes will only take effect if you click save)

Figure 4.2.5: Screenshot of the OpenMRS MDSS homepage.
4.3 Testing

Although the MDSS project did not directly involve handling any tickets in OpenMRS trunk itself, the time spent getting a running development environment of OpenMRS was useful for installing the MDS module and generating test packages for use in the MDSS.

The MDSS was successfully tested for graceful failure behavior on upload with different input files including: a) valid packages, b) zip archives with corrupted XML files, c) zip archives with no xml files, d) corrupted zip files, e) php files, f) images, and g) XML files. The MDSS was tested on 5 browsers: Chrome, Chromium, Firefox, Aurora, and Safari. There is some cross-browser support in the code, especially in download.php (see Section F.13), and in the elements which use JQuery and JQuery UI. Overall cross-browser support needs more consideration in the next generation of MDSS.

For code formatting, several different external checkers were used. HTML output of the MDSS is run through the W3Schools -XHTML standard checker to maintain well-formed output. Plain Javascript code (for example, see Section F.23 in Appendix F) is run through JSLint style checker. JSON output is run through JSONLint validator. Analogous to the rigorous unit tests of the OpenMRS Java coding guidelines, the simpletest.org PHP unit testing framework was investigated for unit testing. Simpletest has very good support for mocking classes and doing test driven design. However, this MDSS design was already set by the time Simpletest became an option and it is not Object Oriented enough for the tests to be pervasive. Instead all functions were independently tested to see that they fulfill their interface contracts. This implementation of the MDSS was not stress-tested and did not undergo any non-technical user trials.
Chapter 5

Evaluation

The original success evaluation criteria for this project proposed in October were: 1) OpenMRS successfully using the MDSS software, 2) MDSS integrated into MDS Module, 3) project keeping on schedule, and 4) intangibles of experience/learning with real world software engineering and development.

Two of those original criteria have been abandoned as failed along the way. The initial schedule was not kept, and although it might be shown to be unrealistic, I made a case that it was okay when I proposed it and it would not have been approved if it was entirely unfeasible. Nonetheless, it had to be abandoned because the milestones were consistently not met. With the shift from Java to PHP, the potential integration with the MDS module was also delayed indefinitely. The MDSS and MDS can still work together well by sharing packages as zip archives, but they still require minimal user action to move the files from one to the other.

The other two original criteria have been successfully met. OpenMRS has not had time yet to use the MDSS in production, but my mentor Rafal successfully deployed the MDSS onto XAMPP on the OpenMRS server on May 2nd. His feedback on the code is that it is “clean and easy to understand” and he also stated his opinion that the logic of the code successfully met the functionality goals for the server although the UI was not what he had envisioned. With notable caveats he even thought it could be ‘good’. Those caveats include the following changes: 1) refactoring to separate logic from view, 2) moving the PHP into a framework such as Zend or Cake to make further UI development easier for someone with CSS-only skills, and 3) switching the database calls over to an ORM. The MDSS is the subject of a Google Summer of Code Project this summer and this code may
or may not find a home in the next generation of server. The code in Appendix F is in flux even now as I improve the MDSS with new ideas that did not make it into the build that was deployed. This is the nature of OSS work, that all contributions go into the code base and will likely be adapted or co-opted by future developers, but there is no need for egos when all improvements are helping a humanitarian cause. When it comes to intangibles this project has bought lots of experience. I have new appreciation and a closer perspective on the difficulties of project scheduling and maintaining working relationships under distributed development and long distance collaboration.

I would also like to suggest a new fifth success criteria for this project. Any software project following engineering processes and generating a formal requirements specification, will be validated against those same requirements. There is no formal validation process in place for this project, but it has succeeded in meeting all of the points of its SRS. A success criteria of functionality, meeting the elicited and specified requirements, is met in this project. Overall, my project meets at least 3 out of 5 of its success criteria, and has both benefited the world and hopefully set a pattern for more student work to do so.
Chapter 6

Conclusion

6.1 Future Work

In the short term there are several features that occurred to me during the development of this implementation of the MDSS which given another month of work I might like to include. Some of these additions include: a) allowing uploading of password protected or private packages, b) other increased security measures such as password strength requirements and better facility for users to recover their own lost or forgotten password, c) comments on packages, d) tracking of time between user logins for inactivity, e) a warning/banning system to allow removal of a users uploading privileges without removing them from the server entirely, and f) a complete style/theme for the page instead of random CSS hacks.

In the longer term future, given another semester or two to continue working on the MDSS there are many more options to consider and directions that I would take. I want to expand on the proofs of concept featured in this implementation which features examples of many techniques that could be further expanded to improve the user experience and capabilities of the MDSS. Javascript, which was used for some form validation, searching, and tabbed display, is the technology with the most potential for upgrading the MDSS. Especially interesting is the potential further use of a javascript framework for this task. A PHP framework such as Cake could also be investigated. Chances are that something general that I would implement has been implemented by someone else before and been standardized, made more full featured, efficient, and/or elegant, and generally improved for inclusion in a framework library. JQuery and JQuery UI are both Javascript frameworks which have lots to offer the MDSS and JQuery UI in particular might be useful in composing a unifying
theme and style for the whole UI. Because this server was targeted to use a MySQL database it is coded directly with MySQL RDBMS in mind. A good expansion in the future would be to replace this with the abstraction of an ORM system such as PDO so it can be storage solution independent.

Finally, a major design pattern that I learned about through the community at OpenMRS but was too far along to re-implement to use at this time is RESTful web services patterns. These web services define four design principles that greatly neaten up and formalize the interfaces of the server: 1) Use HTTP methods (Requests) explicitly, 2) Expose directory structure-like URIs, 3) be Stateless, and 4) Transfer XML, JavaScript Object Notation (JSON), or both. Implementation of RESTful style web services would provide a much cleaner, orderly design to the MDSS as well as helping aid further integration with the main OpenMRS platform and the MDS module in particular. The ideal end goal is for the MDS module to interact with the MDSS directly so that the lists of uploaded packages are visible in the MDS and no extra import of the downloaded packages from file is necessary because they download directly into the MDS import processes.

6.2 Conclusion

The OpenMRS MDSS project was a very good introduction to a world of software development which the curriculum at Princeton University does not focus on. Connections were made between faculty and developers, and hopefully the particular complications of this project have not hampered future academic-HFOSS collaborative endeavors. It is my hope that, having proven the feasibility of independent work in a real world project that helps others, it will inspire future students to also find their interest in making the world a better place.
Bibliography


Appendix A

Global OpenMRS Distribution

Figure A.0.1: A map of current OpenMRS clinical and research sites
<table>
<thead>
<tr>
<th>Type</th>
<th>Locality</th>
<th>Country</th>
<th>Site</th>
<th>Latitude</th>
<th>Longitude</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clinical</td>
<td>Buenos Aires</td>
<td>Argentina</td>
<td>Gobierno de la Ciudad de Buenos Aires</td>
<td>-34.5761</td>
<td>-58.4068</td>
</tr>
<tr>
<td>Research</td>
<td>Yerevan</td>
<td>Armenia</td>
<td>American University of Armenia</td>
<td>40.193922</td>
<td>44.503964</td>
</tr>
<tr>
<td>Clinical</td>
<td>Palacios</td>
<td>Bolivia</td>
<td>Centro Médico Humberto Parra</td>
<td>-17.35</td>
<td>-63.6</td>
</tr>
<tr>
<td>Research</td>
<td>Santa Maria</td>
<td>Brasil</td>
<td>Universidade Federal de Santa Maria</td>
<td>-29.712087</td>
<td>-53.7167</td>
</tr>
<tr>
<td>Clinical</td>
<td>Siem Reap</td>
<td>Cambodia</td>
<td>Angkor Hospital for Children</td>
<td>13.345</td>
<td>103.854</td>
</tr>
<tr>
<td>Clinical</td>
<td>Siem Reap</td>
<td>Cambodia</td>
<td>Siem Reap Provincial Hospital of Cambodia</td>
<td>13.36865</td>
<td>103.86441</td>
</tr>
<tr>
<td>Clinical</td>
<td>Douala</td>
<td>Cameroon</td>
<td>Hôpital Général De Douala</td>
<td>4.0636</td>
<td>9.7589</td>
</tr>
<tr>
<td>Research</td>
<td>Santiago</td>
<td>Chile</td>
<td></td>
<td>-33.425362</td>
<td>-70.566467</td>
</tr>
<tr>
<td>Research</td>
<td>Temuco</td>
<td>Chile</td>
<td></td>
<td>-38.7333</td>
<td>-72.6</td>
</tr>
<tr>
<td>Research</td>
<td>Beijing</td>
<td>China</td>
<td>Beijing University of Aeronautics and Astronautics</td>
<td>39.943</td>
<td>116.388</td>
</tr>
<tr>
<td>Clinical</td>
<td>Brazzaville</td>
<td>Congo</td>
<td>CANTAM Project : Center for Vegetable Resources Studies</td>
<td>-4.2592</td>
<td>15.2847</td>
</tr>
<tr>
<td>Research</td>
<td>Camagüey</td>
<td>Cuba</td>
<td>University of Camagüey</td>
<td>21.386</td>
<td>-77.914</td>
</tr>
<tr>
<td>Clinical</td>
<td>Koraro</td>
<td>Ethiopia</td>
<td>Millennium Villages Project</td>
<td>13.978</td>
<td>39.43</td>
</tr>
<tr>
<td>Clinical</td>
<td>Lambaréné</td>
<td>Gabon</td>
<td>Albert Schweitzer Hospital</td>
<td>-0.701546</td>
<td>10.240386</td>
</tr>
<tr>
<td>Research</td>
<td>Göttingen</td>
<td>Germany</td>
<td>University of Göttingen</td>
<td>51.53404</td>
<td>9.938464</td>
</tr>
<tr>
<td>Research</td>
<td>Freiburg im Breisgau</td>
<td>Germany</td>
<td>University of Freiburg</td>
<td>48.119</td>
<td>8.399</td>
</tr>
<tr>
<td>Research</td>
<td>Acerca</td>
<td>Ghana</td>
<td>Ghana Health Service</td>
<td>5.555717</td>
<td>-0.196306</td>
</tr>
<tr>
<td>Clinical</td>
<td>Bonsasso</td>
<td>Ghana</td>
<td>Millennium Villages Project</td>
<td>6.687119</td>
<td>-1.621969</td>
</tr>
<tr>
<td>Research</td>
<td>Guatemala City</td>
<td>Guatemala</td>
<td>Universidad del Valle de Guatemala</td>
<td>14.643</td>
<td>-90.513</td>
</tr>
<tr>
<td>Clinical</td>
<td>Deschapelles</td>
<td>Haiti</td>
<td>Albert Schweitzer Hospital</td>
<td>19.083332</td>
<td>-72.5</td>
</tr>
<tr>
<td>Clinical</td>
<td>Port-au-Prince</td>
<td>Haiti</td>
<td>Matissant Health Center</td>
<td>18.539268</td>
<td>-72.336411</td>
</tr>
<tr>
<td>Clinical</td>
<td>Cange</td>
<td>Haiti</td>
<td>Partners In Health</td>
<td>18.93498</td>
<td>-71.94965</td>
</tr>
<tr>
<td>Clinical</td>
<td>Petite-Riviere-de-Nippe</td>
<td>Haiti</td>
<td>Visitation Hospital</td>
<td>18.475124</td>
<td>-73.2304</td>
</tr>
<tr>
<td>Clinical</td>
<td>Nuevo Paraíso</td>
<td>Honduras</td>
<td>Global Brigades</td>
<td>13.86414</td>
<td>-86.552689</td>
</tr>
<tr>
<td>Clinical</td>
<td>Dehradun</td>
<td>India</td>
<td>Doon Hospital</td>
<td>30.33357</td>
<td>78.04776</td>
</tr>
<tr>
<td>Research</td>
<td>Kuningan</td>
<td>Indonesia</td>
<td>University of Bakrie School of Management</td>
<td>-6.220711</td>
<td>106.833618</td>
</tr>
<tr>
<td>Research</td>
<td>Jerusalem</td>
<td>Israel</td>
<td>Hebrew University of Jerusalem</td>
<td>31.779</td>
<td>35.2253</td>
</tr>
<tr>
<td>Research</td>
<td>Bolzano</td>
<td>Italy</td>
<td>Free University of Bozen</td>
<td>46.4927</td>
<td>11.3336</td>
</tr>
<tr>
<td>Clinical</td>
<td>Tokushima</td>
<td>Japan</td>
<td>University of Tokushima Medical and Dental Hospital</td>
<td>34.0667</td>
<td>134.5667</td>
</tr>
<tr>
<td>Clinical</td>
<td>Eldoret</td>
<td>Kenya</td>
<td>AMPATH</td>
<td>0.51524</td>
<td>35.266369</td>
</tr>
<tr>
<td>Clinical</td>
<td>Kisumu</td>
<td>Kenya</td>
<td>Lumumba Health Centre</td>
<td>-0.09516</td>
<td>34.74733</td>
</tr>
<tr>
<td>Clinical</td>
<td>Garissa</td>
<td>Kenya</td>
<td>Millennium Villages Project</td>
<td>-0.444659</td>
<td>39.671711</td>
</tr>
<tr>
<td>Clinical</td>
<td>Sauri</td>
<td>Kenya</td>
<td>Millennium Villages Project</td>
<td>0.10131</td>
<td>34.200765</td>
</tr>
<tr>
<td>Clinical</td>
<td>Nohama</td>
<td>Lesotho</td>
<td>Partners In Health</td>
<td>-30.054213</td>
<td>27.854214</td>
</tr>
<tr>
<td>Clinical</td>
<td>Bela-Bela</td>
<td>Lesotho</td>
<td>Monyetla Centre</td>
<td>-29.02285</td>
<td>28.01702</td>
</tr>
<tr>
<td>Clinical</td>
<td>Lilongwe</td>
<td>Malawi</td>
<td>Baobab Health Partnership</td>
<td>-13.9826</td>
<td>33.77362</td>
</tr>
<tr>
<td>Clinical</td>
<td>Neno</td>
<td>Malawi</td>
<td>Partners In Health</td>
<td>-15.55976</td>
<td>34.504368</td>
</tr>
<tr>
<td>Clinical</td>
<td>Namibete</td>
<td>Malawi</td>
<td>St. Gabriël's Hospital</td>
<td>-14.0167</td>
<td>33.35</td>
</tr>
<tr>
<td>Clinical</td>
<td>Sabah</td>
<td>Malaysia</td>
<td>Tenom Hospital</td>
<td>5.5</td>
<td>117</td>
</tr>
<tr>
<td>Clinical</td>
<td>Ségou</td>
<td>Mali</td>
<td>Millennium Villages Project</td>
<td>13.440597</td>
<td>-6.265322</td>
</tr>
<tr>
<td>Clinical</td>
<td>Tíby</td>
<td>Mali</td>
<td>Millennium Villages Project</td>
<td>14.116667</td>
<td>-7.983333</td>
</tr>
<tr>
<td>Clinical</td>
<td>Toya</td>
<td>Mali</td>
<td>Millennium Villages Project</td>
<td>12.342</td>
<td>-7.869</td>
</tr>
<tr>
<td>Clinical</td>
<td>Souraрабougou</td>
<td>Mali</td>
<td>Mali Health Organizing Project</td>
<td>13.6552</td>
<td>-9.0256</td>
</tr>
<tr>
<td>Clinical</td>
<td>Mozambique</td>
<td>MOASIS - South African Medical Research Council</td>
<td>-25.968945</td>
<td>32.56955</td>
<td></td>
</tr>
<tr>
<td>Clinical</td>
<td>Nepal</td>
<td>Nepal</td>
<td>World Health Organization</td>
<td>27.702871</td>
<td>85.318245</td>
</tr>
<tr>
<td>Clinical</td>
<td>Sanepa</td>
<td>Nepal</td>
<td>Star Hospital</td>
<td>27.68421</td>
<td>85.30108</td>
</tr>
<tr>
<td>Clinical</td>
<td>Managua</td>
<td>Nicaragua</td>
<td>Sustainable Sciences Institute</td>
<td>12.136</td>
<td>-86.275</td>
</tr>
<tr>
<td>Research</td>
<td>Managua</td>
<td>Nicaragua</td>
<td>Sustainable Sciences Institute</td>
<td>12.136</td>
<td>-86.275</td>
</tr>
<tr>
<td>Clinical</td>
<td>Ikaram</td>
<td>Nigeria</td>
<td>Millennium Villages Project</td>
<td>7.1</td>
<td>4.833</td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td>---------</td>
<td>------------------------------</td>
<td>-----</td>
<td>-------</td>
</tr>
<tr>
<td>Clinical</td>
<td>Pampaida</td>
<td>Nigeria</td>
<td>Millennium Villages Project</td>
<td>10.523</td>
<td>7.44</td>
</tr>
<tr>
<td>Clinical</td>
<td>Kaduna</td>
<td>Nigeria</td>
<td>Shehu Idris College of Health Sciences and Technology</td>
<td>10.51901</td>
<td>7.443122</td>
</tr>
<tr>
<td>Clinical</td>
<td>Karachi</td>
<td>Pakistan</td>
<td>Indus Hospital Research Center</td>
<td>24.893379</td>
<td>67.028061</td>
</tr>
<tr>
<td>Clinical</td>
<td>Karachi</td>
<td>Pakistan</td>
<td></td>
<td>24.893379</td>
<td>67.028061</td>
</tr>
<tr>
<td>Research</td>
<td>Karachi</td>
<td>Pakistan</td>
<td></td>
<td>24.893379</td>
<td>67.028061</td>
</tr>
<tr>
<td>Clinical</td>
<td>Lima</td>
<td>Peru</td>
<td>Socios En Salud Sucursal Peru</td>
<td>-12.093084</td>
<td>-77.046494</td>
</tr>
<tr>
<td>Clinical</td>
<td>Santa Anita</td>
<td>Peru</td>
<td>Hospital Hermilio Valdizan</td>
<td>-12.0333</td>
<td>-76.9667</td>
</tr>
<tr>
<td>Clinical</td>
<td>Lima</td>
<td>Peru</td>
<td>Hospital Daniel A. Carrion</td>
<td>-12.06016</td>
<td>-77.12797</td>
</tr>
<tr>
<td>Clinical</td>
<td>Manila</td>
<td>Philippines</td>
<td>National Chronic Disease Registry - Philippine Diabetes Registry</td>
<td>14.601033</td>
<td>120.976158</td>
</tr>
<tr>
<td>Research</td>
<td>Manila</td>
<td>Philippines</td>
<td>University of the Philippines National Telehealth Center</td>
<td>14.601033</td>
<td>120.976158</td>
</tr>
<tr>
<td>Research</td>
<td>Legazpi</td>
<td>Philippines</td>
<td>Bicol University</td>
<td>13.5</td>
<td>123.5</td>
</tr>
<tr>
<td>Medical</td>
<td>Mayange</td>
<td>Rwanda</td>
<td>Millennium Villages Project</td>
<td>-2.2114</td>
<td>30.1172</td>
</tr>
<tr>
<td>Medical</td>
<td>Rwinkwavu</td>
<td>Rwanda</td>
<td>Partners In Health</td>
<td>-1.902265</td>
<td>30.509197</td>
</tr>
<tr>
<td>Medical</td>
<td>Kigali</td>
<td>Rwanda</td>
<td>Partners In Health, TB program, MOH</td>
<td>-1.950106</td>
<td>30.058769</td>
</tr>
<tr>
<td>Research</td>
<td>Kigali</td>
<td>Rwanda</td>
<td>TRACplus Clinic</td>
<td>-1.950106</td>
<td>30.058769</td>
</tr>
<tr>
<td>Clinical</td>
<td>Gitwe</td>
<td>Rwanda</td>
<td>Moi + Medical Missions for Children</td>
<td>-2.2461</td>
<td>29.689301</td>
</tr>
<tr>
<td>Research</td>
<td>Jeddah</td>
<td>Saudi Arabia</td>
<td>Capitals Institute of Health Sciences</td>
<td>21.5169</td>
<td>39.2192</td>
</tr>
<tr>
<td>Clinical</td>
<td>Potou</td>
<td>Senegal</td>
<td>Millennium Villages Project</td>
<td>15.720919</td>
<td>16.506308</td>
</tr>
<tr>
<td>Clinical</td>
<td>Freetown</td>
<td>Sierra Leone</td>
<td>Connaught Hospital</td>
<td>8.48799</td>
<td>-13.23829</td>
</tr>
<tr>
<td>Research</td>
<td>Umkhanyakude</td>
<td>South Africa</td>
<td>Africa Centre PCIS</td>
<td>-27.616667</td>
<td>32.033333</td>
</tr>
<tr>
<td>Clinical</td>
<td>Cape Town</td>
<td>South Africa</td>
<td>Desmond Tutu HIV Center</td>
<td>-33.923775</td>
<td>18.432346</td>
</tr>
<tr>
<td>Research</td>
<td>Cape Town</td>
<td>South Africa</td>
<td>Jembi Headoffice</td>
<td>-33.923775</td>
<td>18.432346</td>
</tr>
<tr>
<td>Research</td>
<td>Durban</td>
<td>South Africa</td>
<td>SA Medical Research Council/UKZN</td>
<td>-29.857002</td>
<td>31.024794</td>
</tr>
<tr>
<td>Research</td>
<td>Bloemfontein, Free State</td>
<td>South Africa</td>
<td>University of the Free State Department of Virology</td>
<td>-29.116</td>
<td>26.15</td>
</tr>
<tr>
<td>Research</td>
<td>Pretoria</td>
<td>South Africa</td>
<td>University of South Africa</td>
<td>-25.7069</td>
<td>28.2294</td>
</tr>
<tr>
<td>Clinical</td>
<td>Tygerberg</td>
<td>South Africa</td>
<td>Tygerberg Hospital</td>
<td>-33.910433</td>
<td>18.613179</td>
</tr>
<tr>
<td>Clinical</td>
<td>Queenstown</td>
<td>South Africa</td>
<td>Frontier Hospital</td>
<td>-31.9817</td>
<td>26.8819</td>
</tr>
<tr>
<td>Research</td>
<td>Colombo</td>
<td>Sri Lanka</td>
<td>University of Colombo Software Development Unit</td>
<td>6.901948</td>
<td>79.861561</td>
</tr>
<tr>
<td>Clinical</td>
<td>Colombo</td>
<td>Sri Lanka</td>
<td>Post Graduate Institute of Medicine</td>
<td>6.917523</td>
<td>79.869306</td>
</tr>
<tr>
<td>Research</td>
<td>Peradeniya</td>
<td>Sri Lanka</td>
<td>University of Peradeniya Department of Computer Engineering</td>
<td>7.253823</td>
<td>80.5917</td>
</tr>
<tr>
<td>Research</td>
<td>Geneva</td>
<td>Switzerland</td>
<td>World Health Organization (WHO)</td>
<td>46.2022</td>
<td>6.1457</td>
</tr>
<tr>
<td>Research</td>
<td>Geneva</td>
<td>Switzerland</td>
<td>University of Geneva</td>
<td>46.194887</td>
<td>6.140055</td>
</tr>
<tr>
<td>Medical</td>
<td>Mbola</td>
<td>Tanzania</td>
<td>Million Villages Project</td>
<td>-4.9167</td>
<td>32.8333</td>
</tr>
<tr>
<td>Clinical</td>
<td>Kiraha</td>
<td>Tanzania</td>
<td>University of Dar es Salaam</td>
<td>-6.7667</td>
<td>38.916698</td>
</tr>
<tr>
<td>Clinical</td>
<td>Dar es Salaam</td>
<td>Tanzania</td>
<td>University of Dar es Salaam</td>
<td>-6.822921</td>
<td>39.269661</td>
</tr>
<tr>
<td>Clinical</td>
<td>Morogoro</td>
<td>Tanzania</td>
<td>University of Dar es Salaam</td>
<td>-6.822097</td>
<td>37.6616</td>
</tr>
<tr>
<td>Research</td>
<td>Bangkok</td>
<td>Thailand</td>
<td>International School of Bangkok</td>
<td>13.75</td>
<td>100.5167</td>
</tr>
<tr>
<td>Clinical</td>
<td>Fajara, Banjul</td>
<td>The Gambia</td>
<td>Medical Research Council (UK) The Gambia</td>
<td>13.4716</td>
<td>-16.6968</td>
</tr>
<tr>
<td>Clinical</td>
<td>Kampala</td>
<td>Uganda</td>
<td>Makerere University</td>
<td>0.314269</td>
<td>32.572872</td>
</tr>
<tr>
<td>Clinical</td>
<td>Ruhirwa</td>
<td>Uganda</td>
<td>Millennium Villages Project</td>
<td>-0.843543</td>
<td>30.803947</td>
</tr>
<tr>
<td>Clinical</td>
<td>Mbarara</td>
<td>Uganda</td>
<td>Immune Suppression Syndrome (ISS) Clinic</td>
<td>-0.605675</td>
<td>30.64855</td>
</tr>
<tr>
<td>Clinical</td>
<td>Masaka</td>
<td>Uganda</td>
<td>Masaka Regional Referral Hospital</td>
<td>-0.3128</td>
<td>31.7131</td>
</tr>
<tr>
<td>Clinical</td>
<td>Kasene</td>
<td>Uganda</td>
<td>Kagando Hospital</td>
<td>0.23</td>
<td>29.9883</td>
</tr>
<tr>
<td>Clinical</td>
<td>Masaka</td>
<td>Uganda</td>
<td>Kitovo Mobile AIDS Organisation</td>
<td>0.5</td>
<td>31.75</td>
</tr>
<tr>
<td>Research</td>
<td>Leeds</td>
<td>United Kingdom</td>
<td>University of Leeds</td>
<td>53.801335</td>
<td>-1.553278</td>
</tr>
<tr>
<td>Research</td>
<td>Southampton</td>
<td>United Kingdom</td>
<td>University of Southampton School of Health Sciences</td>
<td>50.92872</td>
<td>-1.40266</td>
</tr>
<tr>
<td>Clinical</td>
<td>Indianapolis, Indiana</td>
<td>United States</td>
<td>Child Health Improvement through Computer Automation (CHICA)</td>
<td>39.77726</td>
<td>-86.18008</td>
</tr>
<tr>
<td>Clinical</td>
<td>Indianapolis, Indiana</td>
<td>United States</td>
<td>Indianapolis Motor Speedway</td>
<td>39.790975</td>
<td>-86.235133</td>
</tr>
<tr>
<td>Research</td>
<td>Indianapolis, Indiana</td>
<td>United States</td>
<td>OpenMRS LLC</td>
<td>39.78133</td>
<td>-86.16678</td>
</tr>
<tr>
<td>Research</td>
<td>Boston, Massachusetts</td>
<td>United States</td>
<td>Partners In Health</td>
<td>42.358433</td>
<td>-71.059776</td>
</tr>
<tr>
<td>Research</td>
<td>Indianapolis, Indiana</td>
<td>United States</td>
<td>Regenstrief Institute</td>
<td>39.77864</td>
<td>-86.17785</td>
</tr>
<tr>
<td>Clinical</td>
<td>Los Angeles, California</td>
<td>United States</td>
<td>Skid Row Homeless Healthcare Initiative</td>
<td>34.052235</td>
<td>-118.24368</td>
</tr>
<tr>
<td>Research/Legal Site</td>
<td>City, State</td>
<td>Country</td>
<td>Organization</td>
<td>Lat.</td>
<td>Long.</td>
</tr>
<tr>
<td>---------------------</td>
<td>-------------</td>
<td>---------</td>
<td>--------------</td>
<td>------</td>
<td>-------</td>
</tr>
<tr>
<td>Oak Hill, Virginia</td>
<td>United States</td>
<td>SYSNET International, Inc.</td>
<td>38.937763</td>
<td>-77.412819</td>
<td></td>
</tr>
<tr>
<td>Pittsburgh, Pennsylvania</td>
<td>United States</td>
<td>University of Pittsburgh</td>
<td>40.443184</td>
<td>-79.961586</td>
<td></td>
</tr>
<tr>
<td>Carrollton, Georgia</td>
<td>United States</td>
<td>University of West Georgia</td>
<td>33.580109</td>
<td>-85.076614</td>
<td></td>
</tr>
<tr>
<td>Nashville, Tennessee</td>
<td>United States</td>
<td>Vanderbilt University</td>
<td>36.142952</td>
<td>-86.805489</td>
<td></td>
</tr>
<tr>
<td>San Mateo, California</td>
<td>United States</td>
<td>Actuate Corporation</td>
<td>37.563</td>
<td>-122.3255</td>
<td></td>
</tr>
<tr>
<td>Palo Alto, California</td>
<td>United States</td>
<td>Stanford University School of Medicine</td>
<td>37.4312</td>
<td>-122.1703</td>
<td></td>
</tr>
<tr>
<td>Denver, Colorado</td>
<td>United States</td>
<td>Genesee Academy LLC</td>
<td>39.739</td>
<td>-104.985</td>
<td></td>
</tr>
<tr>
<td>Salina, Kansas</td>
<td>United States</td>
<td>Kansas Wesleyan University</td>
<td>38.813892</td>
<td>-97.609478</td>
<td></td>
</tr>
<tr>
<td>Boston, Massachusetts</td>
<td>United States</td>
<td>Harvard Medical School</td>
<td>42.3659</td>
<td>-71.1226</td>
<td></td>
</tr>
<tr>
<td>Boston, Massachusetts</td>
<td>United States</td>
<td>Children's Hospital Boston</td>
<td>42.337126</td>
<td>-71.106037</td>
<td></td>
</tr>
<tr>
<td>Kansas City, Missouri</td>
<td>United States</td>
<td>Children International</td>
<td>39.1</td>
<td>-94.581</td>
<td></td>
</tr>
<tr>
<td>Tulsa, Oklahoma</td>
<td>United States</td>
<td>Tulsa Community College</td>
<td>36.04384</td>
<td>-95.86203</td>
<td></td>
</tr>
<tr>
<td>Portland, Oregon</td>
<td>United States</td>
<td>Oregon Health &amp; Science University</td>
<td>45.524</td>
<td>-122.668</td>
<td></td>
</tr>
<tr>
<td>Conshohocken, Pennsylvania</td>
<td>United States</td>
<td>ePharmaSolutions</td>
<td>40.077</td>
<td>-75.303</td>
<td></td>
</tr>
<tr>
<td>Pittsburgh, Pennsylvania</td>
<td>United States</td>
<td>University of Pittsburgh</td>
<td>40.444752</td>
<td>-79.952568</td>
<td></td>
</tr>
<tr>
<td>Philadelphia, Pennsylvania</td>
<td>United States</td>
<td>The Children's Hospital of Philadelphia</td>
<td>39.94816</td>
<td>-75.19464</td>
<td></td>
</tr>
<tr>
<td>Charlottesville, Virginia</td>
<td>United States</td>
<td>University of Virginia</td>
<td>38.060495</td>
<td>-78.502842</td>
<td></td>
</tr>
<tr>
<td>Seattle, Washington</td>
<td>United States</td>
<td>University of Washington</td>
<td>47.656099</td>
<td>-122.31055</td>
<td></td>
</tr>
<tr>
<td>Harare</td>
<td>Zimbabwe</td>
<td>OASIS - South African Medical Research Council</td>
<td>-17.82922</td>
<td>31.053961</td>
<td></td>
</tr>
<tr>
<td>Chitungwiza</td>
<td>Zimbabwe</td>
<td>Opportunistic Infection Clinic</td>
<td>-18.01978</td>
<td>31.067907</td>
<td></td>
</tr>
</tbody>
</table>

Figure A.0.2: A table listing the current OpenMRS clinical and research sites
## Appendix B

### Initial Project Schedule

Joseph Wilder Metadata Sharing Server : Initial Project Plan  
Start Date: 10/27/11  End Date: 04/30/12

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
<th>Start Date</th>
<th>Duration</th>
<th>Completion Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server</td>
<td>1.1 Project definition tasks</td>
<td>10/27/11</td>
<td>53</td>
<td>12/20/11</td>
</tr>
<tr>
<td></td>
<td>1.2 Learning tasks (Eclipse, Java servers, threads, Server)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.3 Scenarios and Requirements (SRS)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.4 GUI mock-ups for uploading file, listing and</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.5 Development Environment</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.6 Server Implementation – web interface</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.7 Documentation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feature Pack 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.1 Integration with Metadata Sharing Module</td>
<td>12/27/11</td>
<td>38</td>
<td>02/05/12</td>
<td></td>
</tr>
<tr>
<td>2.2 Scenarios and Requirements (SRS)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.3 Authentication of Publishers</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.4 GUI mock-ups for uploading, downloading</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.5 Implementation in server and MDSM</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.6 Documentation</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feature Pack 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.1 Scenarios and Requirements (SRS)</td>
<td>02/10/12</td>
<td>29</td>
<td>03/11/12</td>
<td></td>
</tr>
<tr>
<td>3.2 GUI mock-ups for search, stats, recommendation?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.3 Implementation in server and MDSM</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.4 Documentation</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Updates from Feedback</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.1 Open tickets in JIRA</td>
<td>03/12/12</td>
<td>19</td>
<td>03/31/12</td>
<td></td>
</tr>
<tr>
<td>4.2 Other feedback</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.3 Documentation</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Documentation</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.1 Write documentation and university paper</td>
<td>03/31/12</td>
<td>12</td>
<td>04/12/12</td>
<td></td>
</tr>
</tbody>
</table>

Figure B.0.1: The initial project management schedule.
Appendix C

Example Metadata Package

C.1 header.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<package id="1" uuid="f446e1b9-10ca-4e2a-822f-ad0c972158fb"/>
<dateCreated id="2">2011-12-02 15:59:40</dateCreated>
<name>New NAME</name>
<description>Test package 1 of ?.</description>
<openmrsVersion>1.9.0-SNAPSHOT</openmrsVersion>
<version>1</version>
<group>3bc4ab33-18e8-4e5a-817b-0b08a13bb9e1</group>
<subscriptionUrl>http://localhost:8080/openmrs/ws/rest/metadataSharing/package/3bc4ab33-18e8-4e5a-817b-0b08a13bb9e1/latest</subscriptionUrl>
<items id="3">
    <explicitItem id="4">
        <classname>org.openmrs.Role</classname>
        <uuid>8d94f280-c2cc-11de-8d13-0010c6dffd0f</uuid>
        <name>Provider</name>
    </explicitItem>
    <explicitItem id="5">
        <classname>org.openmrs.Role</classname>
        <uuid>f7fd42ef-880c-40c5-972d-e4ae7e990de2</uuid>
        <name>Authenticated</name>
    </explicitItem>
    <explicitItem id="6">
        <classname>org.openmrs.Role</classname>
        <uuid>774b2af3-6437-4e5a-a310-547554c7e65c</uuid>
        <name>Anonymous</name>
    </explicitItem>
    <explicitItem id="7">
        <classname>org.openmrs.Role</classname>
        <uuid>8d94f852-c2cc-11de-8d13-0010c6dffd0f</uuid>
        <name>System Developer</name>
    </explicitItem>
    <explicitItem id="8">
        <classname>org.openmrs.Role</classname>
        <uuid>6a1378ce-1d1d-11e1-94a3-0018de157d2c</uuid>
        <name>Clinician</name>
    </explicitItem>
    <explicitItem id="9">
        <classname>org.openmrs.Role</classname>
        <uuid>6a137bda-1d1d-11e1-94a3-0018de157d2c</uuid>
        <name>Data Assistant</name>
    </explicitItem>
    <explicitItem id="10">
        <classname>org.openmrs.Role</classname>
        <uuid>6a1383b0-1d1d-11e1-94a3-0018de157d2c</uuid>
        <name>Data Manager</name>
    </explicitItem>
</items>
```

<name>View Problems</name>
</includedDependency>

<includedDependency id="12">
  <classname>org.openmrs.Privilege</classname>
  <uuid>68dc5301-1d1d-11e1-94a3-0018de157d2c</uuid>
  <name>View Concepts</name>
</includedDependency>

<includedDependency id="13">
  <classname>org.openmrs.Privilege</classname>
  <uuid>68dc5917-1d1d-11e1-94a3-0018de157d2c</uuid>
  <name>View Encounters</name>
</includedDependency>

<includedDependency id="14">
  <classname>org.openmrs.Privilege</classname>
  <uuid>68dc6f5d-1d1d-11e1-94a3-0018de157d2c</uuid>
  <name>View Patient Identifiers</name>
</includedDependency>

<includedDependency id="15">
  <classname>org.openmrs.Privilege</classname>
  <uuid>68dc3937-1d1d-11e1-94a3-0018de157d2c</uuid>
  <name>View Allergies</name>
</includedDependency>

<includedDependency id="16">
  <classname>org.openmrs.Privilege</classname>
  <uuid>68dc673f-1d1d-11e1-94a3-0018de157d2c</uuid>
  <name>View Observations</name>
</includedDependency>

<includedDependency id="17">
  <classname>org.openmrs.Privilege</classname>
  <uuid>68dc737a-1d1d-11e1-94a3-0018de157d2c</uuid>
  <name>View Patients</name>
</includedDependency>

<includedDependency id="18">
  <classname>org.openmrs.Privilege</classname>
  <uuid>68dc757d-1d1d-11e1-94a3-0018de157d2c</uuid>
  <name>View People</name>
</includedDependency>

<includedDependency id="19">
  <classname>org.openmrs.Privilege</classname>
  <uuid>68dc5f20-1d1d-11e1-94a3-0018de157d2c</uuid>
  <name>View Global Properties</name>
</includedDependency>

<includedDependency id="20">
  <classname>org.openmrs.Privilege</classname>
  <uuid>68dc87d5-1d1d-11e1-94a3-0018de157d2c</uuid>
  <name>View Roles</name>
</includedDependency>

<includedDependency id="21">
  <classname>org.openmrs.Privilege</classname>
  <uuid>68dc4ab3-1d1d-11e1-94a3-0018de157d2c</uuid>
  <name>View Concept Classes</name>
</includedDependency>

</items>
</package>
C.2 metadata.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<list id="1">
  <org.openmrs.Role id="2" uuid="8d94f280-c2cc-11de-8d13-0010c6dff0f">
    <description>All users with the 'Provider' role will appear as options in the default Infopath</description>
    <retired>false</retired>
    <role>Provider</role>
    <privileges id="3">
      <org.openmrs.Privilege id="4" uuid="68dc7b8e-1d1d-11e1-94a3-0018de157d2c">
        <description>Able to view problems in OpenMRS</description>
        <retired>false</retired>
        <privilege>View Problems</privilege>
      </org.openmrs.Privilege>
      <org.openmrs.Privilege id="5" uuid="68dc5301-1d1d-11e1-94a3-0018de157d2c">
        <description>Able to view concept entries</description>
        <retired>false</retired>
        <privilege>View Concepts</privilege>
      </org.openmrs.Privilege>
      <org.openmrs.Privilege id="6" uuid="68dc5917-1d1d-11e1-94a3-0018de157d2c">
        <description>Able to view patient encounters</description>
        <retired>false</retired>
        <privilege>View Encounters</privilege>
      </org.openmrs.Privilege>
      <org.openmrs.Privilege id="7" uuid="68dc3937-1d1d-11e1-94a3-0018de157d2c">
        <description>Able to view allergies in OpenMRS</description>
        <retired>false</retired>
        <privilege>View Allergies</privilege>
      </org.openmrs.Privilege>
      <org.openmrs.Privilege id="8" uuid="68dc6f5d-1d1d-11e1-94a3-0018de157d2c">
        <description>Able to view patient identifiers</description>
        <retired>false</retired>
        <privilege>View Patient Identifiers</privilege>
      </org.openmrs.Privilege>
      <org.openmrs.Privilege id="9" uuid="68dc673f-1d1d-11e1-94a3-0018de157d2c">
        <description>Able to view patient observations</description>
        <retired>false</retired>
        <privilege>View Observations</privilege>
      </org.openmrs.Privilege>
      <org.openmrs.Privilege id="10" uuid="68dc737a-1d1d-11e1-94a3-0018de157d2c">
        <description>Able to view patients</description>
        <retired>false</retired>
        <privilege>View Patients</privilege>
      </org.openmrs.Privilege>
      <org.openmrs.Privilege id="11" uuid="68dc757d-1d1d-11e1-94a3-0018de157d2c">
        <description>Able to view person objects</description>
        <retired>false</retired>
        <privilege>View People</privilege>
      </org.openmrs.Privilege>
    </privileges>
    <inheritedRoles id="12"/>
  </org.openmrs.Role>
</list>
```

Princeton University
Privilages gained once authentication has been established.

Authenticated

- Able to view global properties on the administration screen
- Able to view user roles
- Able to view concept classes
- Able to view relationships
- Able to view relationship types
- Able to view patient identifier types
- Able to view concept datatypes
- Able to view field types
- Able to view locations
- Able to view identifier types
- Able to view concept classes
- Able to view concepts
- Able to view field types
- Able to view locations
<org.openmrs.Privilege id="24" uuid="68dc777f-1d1d-11e1-94a3-0018de157d2c">
  <description>Able to view person attribute types</description>
  <retired>false</retired>
  <privilege>View Person Attribute Types</privilege>
</org.openmrs.Privilege>

<org.openmrs.Privilege id="25" uuid="68dc798f-1d1d-11e1-94a3-0018de157d2c">
  <description>Able to view user privileges</description>
  <retired>false</retired>
  <privilege>View Privileges</privilege>
</org.openmrs.Privilege>

<org.openmrs.Privilege id="26" uuid="68dc5710-1d1d-11e1-94a3-0018de157d2c">
  <description>Able to view encounter types</description>
  <retired>false</retired>
  <privilege>View Encounter Types</privilege>
</org.openmrs.Privilege>

<org.openmrs.Privilege id="27" uuid="68dc6944-1d1d-11e1-94a3-0018de157d2c">
  <description>Able to view order types</description>
  <retired>false</retired>
  <privilege>View Order Types</privilege>
</org.openmrs.Privilege>

<org.openmrs.Privilege id="28"/>
</org.openmrs.Roles>

<org.openmrs.Role id="29" uuid="774b2af3-6437-4e5a-a310-547554c7e65c">
  <description>Privileges for non-authenticated users.</description>
  <retired>false</retired>
  <role>Anonymous</role>
  <privileges id="30">
    <org.openmrs.Privilege id="31" uuid="68dc6534-1d1d-11e1-94a3-0018de157d2c">
      <description>Ability to see the navigation menu</description>
      <retired>false</retired>
      <privilege>View Navigation Menu</privilege>
    </org.openmrs.Privilege>
    <inheritedRoles id="32"/>
  </privileges>
</org.openmrs.Role>

<org.openmrs.Role id="33" uuid="8d94f852-c2ce-11de-8d13-0010c6dfff0f">
  <description>Developers of the OpenMRS have additional access to change fundamental structure of the database model.</description>
  <retired>false</retired>
  <role>System Developer</role>
  <privileges id="34"/>
  <inheritedRoles id="35"/>
</org.openmrs.Role>

<org.openmrs.Role id="36" uuid="6a1378ce-1d1d-11e1-94a3-0018de157d2c">
  <description>Privileges for doctors</description>
  <retired>false</retired>
  <role>Clinician</role>
  <privileges id="37"/>
  <inheritedRoles id="38" reference="2"/>
</org.openmrs.Role>

<org.openmrs.Role id="39" uuid="6a137bda-1d1d-11e1-94a3-0018de157d2c">
  <description>Privileges for data-entry operators.</description>
  <retired>false</retired>
</org.openmrs.Role>
<role>Data Assistant</role>

<privileges id="40">
  <org.openmrs.Privilege reference="8"/>
  <org.openmrs.Privilege id="41" uuid="68dbb089-1d1d-11e1-94a3-0018de157d2c">
    <description>Able to add relationships</description>
    <retired>false</retired>
    <privilege>Add Relationships</privilege>
  </org.openmrs.Privilege>
  <org.openmrs.Privilege id="42" uuid="68dc8bdf-1d1d-11e1-94a3-0018de157d2c">
    <description>Able to view users in OpenMRS</description>
    <retired>false</retired>
    <privilege>View Users</privilege>
  </org.openmrs.Privilege>
  <org.openmrs.Privilege id="43" uuid="68dbe9d1-1d1d-11e1-94a3-0018de157d2c">
    <description>Able to edit relationships</description>
    <retired>false</retired>
    <privilege>Edit Relationships</privilege>
  </org.openmrs.Privilege>
  <org.openmrs.Privilege id="44" uuid="68dbcd5b-1d1d-11e1-94a3-0018de157d2c">
    <description>Able to delete relationships</description>
    <retired>false</retired>
    <privilege>Delete Relationships</privilege>
  </org.openmrs.Privilege>
  <org.openmrs.Privilege reference="5"/>
  <org.openmrs.Privilege reference="6"/>
  <org.openmrs.Privilege id="45" uuid="68dc2635-1d1d-11e1-94a3-0018de157d2c">
    <description>Able to view the &apos;Encounters&apos; tab on the patient dashboard</description>
    <retired>false</retired>
    <privilege>Patient Dashboard - View Encounters Section</privilege>
  </org.openmrs.Privilege>
  <org.openmrs.Privilege id="46" uuid="68dc2858-1d1d-11e1-94a3-0018de157d2c">
    <description>Allows user to view the Forms tab on the patient dashboard</description>
    <retired>false</retired>
    <privilege>Patient Dashboard - View Forms Section</privilege>
  </org.openmrs.Privilege>
  <org.openmrs.Privilege id="47" uuid="68dc2417-1d1d-11e1-94a3-0018de157d2c">
    <description>Able to view the &apos;Demographics&apos; tab on the patient dashboard</description>
    <retired>false</retired>
    <privilege>Patient Dashboard - View Demographics Section</privilege>
  </org.openmrs.Privilege>
  <org.openmrs.Privilege id="48" uuid="68dbe56e-1d1d-11e1-94a3-0018de157d2c">
    <description>Able to edit patients</description>
    <retired>false</retired>
    <privilege>Edit Patients</privilege>
  </org.openmrs.Privilege>
</privileges>

<inheritedRoles id="71">
  <org.openmrs.Role reference="39"/>
</inheritedRoles>
</org.openmrs.Role>
</list>
Appendix D

IEEE SRS Standard

IEEE/ANSI 830-1998
Standard structure for Software Requirements Specifications

1. Introduction
   1.1. Purpose of the requirements documentation
   1.2. Scope of the product
   1.3. Definitions, acronyms, and abbreviations
   1.4. References
   1.5. Overview of the remainder of the document

2. General description
   2.1. Product perspective
   2.2. Product functions
   2.3. User Characteristics
   2.4. General Constraints
   2.5. Assumptions and dependencies

3. Specific requirements
   3.1. Functional requirements
   3.2. Non-functional requirements
   3.3. Interface requirements
   3.4. Further structure at the discretion of organisational practice. May include: external interfaces, system functionality and performance, logical database requirements, design constraints, emergent system properties, and quality characteristics.

4. Appendices

5. Index
Database Schema

```sql
-- phpMyAdmin SQL Dump
-- version 3.4.5
-- http://www.phpmyadmin.net

-- Host: localhost
-- Generation Time: May 01, 2012 at 07:20 AM
-- Server version: 5.5.16
-- PHP Version: 5.3.8

SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO"
;
SET time_zone = "+00:00" 
;

Database: 'mdss'

CREATE TABLE IF NOT EXISTS `metadata` (  
`id` int(11) NOT NULL AUTO_INCREMENT,
`name` varchar(255) CHARACTER SET utf8 NOT NULL,
`path` varchar(255) CHARACTER SET utf8 NOT NULL,
`auth` varchar(255) CHARACTER SET utf8 NOT NULL,
`authid` int(11) NOT NULL,
`pubdate` date NOT NULL,
`update` date NOT NULL,
`desc` varchar(510) CHARACTER SET utf8 NOT NULL,
`omrsver` varchar(255) CHARACTER SET utf8 NOT NULL,
`ver` int(11) NOT NULL,
`suburl` varchar(255) CHARACTER SET utf8 NOT NULL,
`dls` int(11) NOT NULL,

PRIMAR Y KEY (`id`) ) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=10 
;

INSERT INTO `metadata` (`id`, `name`, `path`, `auth`, `authid`, `pubdate`, `update`, `desc`, `omrsver`, `ver`, `suburl`, `dls`) VALUES

(6, 'HTML', 'C:/xampp/htdocs/MDSS/uploads/788061
diae339ee8edd85f08d1565b6b7f53b4abda8baef7.zip', 'admin', 4, '2011-12-02', '2012-04-30', 'Test package 1 of 4. ', '1.9.0-SNAPSHOT', 1, 'http://localhost:8080/openmrs/ws/rest/metadatasharing/package/3bc4ab33-18e8-4e5a-817b-0b08a13bb9e1/latest', 1),

(7, 'Espanol', 'C:/xampp/htdocs/MDSS/uploads/4
ee94231e5569a252e70b770653338b9a1077a0c520f.zip', 'Bob', 5, '2011-12-02', '2012-04-30', 'Test package 2 of 4. ', '1.9.0-SNAPSHOT', 1, 'http://localhost:8080/openmrs/ws/rest/metadatasharing/package/3bc4ab33-18e8-4e5a-817b-0b08a13bb9e1/latest', 1),

(8, 'Really Long Name of the Coolest Metadata Package Ever!', 'C:/xampp/htdocs/MDSS/uploads/2550333310861a0a0afdc93964cddbd0eеб48e6920425a.zip', 'a',

```
CREATE TABLE IF NOT EXISTS `users` (  `id` int(11) NOT NULL AUTO_INCREMENT,  `name` varchar(255) CHARACTER SET utf8 NOT NULL,  `email` varchar(255) CHARACTER SET utf8 NOT NULL,  `pw` varchar(255) CHARACTER SET utf8 NOT NULL,  `class` int(11) NOT NULL,  `joindate` date NOT NULL,  `packages` int(11) NOT NULL  ) KEY `id` (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=7;

-- Dumping data for table `users`

INSERT INTO `users` (`id`, `name`, `email`, `pw`, `class`, `joindate`, `packages`) VALUES  (3, 'a', 'a@a.com', '1d5ebced5d3a2042a0f8d830e4fcfaccab50cd3b048dbe1b9', 0, '2012-04-26', 1),  (4, 'admin', 'admin@MDSS.org', '07ef2f6965f5217ea781e042bab327d263a7c1e9e469b93f', 1, '2012-04-26', 1),  (5, 'Bob', 'bob@bob.com', '607b101e41c7757063aced12d73688d6b735124366e99beb', 0, '2012-04-30', 1),  (6, 'test', 'test@test.com', '7698f626e47c73620cc0b827a1bf3f3d3ecc23747e95da76', 1, '2012-04-30', 1);

/*40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
Appendix F

Code

The latest version of this code can be found in the Subversion repository at: https://svn.openmrs.org/openmrs-contrib/metadata-sharing-server/trunk

F.1 Setup

includes.php

```php
<?php
include_once 'configs.php';
include_once 'pwFuncs.php';
include_once 'dbFuncs.php';
include_once 'templates.php';
include_once 'utils.php';
?>
```

cfgins.php

```php
<?php
// path to uploads folder
$uploadpath = dirname(__FILE__) . '/uploads/';

// DB connection details
$dbhost = 'localhost';
$dbuser = 'root';
$dbpass = 'admin';
$dbname = 'MDSS';

// DataTables table
$dttable = 'metadata';
?>
```
<?php
// General template and printing functions
// Basic page start
// Outputs HTML and DOCTYPE, does not handle including includes.php
function startPage($name)
{
    Session_start();
    echo '<!DOCTYPE html PUBLIC "−//W3C//DTD XHTML 1.0 Transitional//EN";'
    echo ' "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">' ;?>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta charset="utf-8"/>
<title>Metadata Sharing Server − <?php echo $name;?></title>
<link rel="stylesheet" type="text/css" href="http://ajax.googleapis.com/ajax/libs/jquui/1.8.9/themes/ui-lightness/jquery-ui.css" />
<script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/1.7.1/jquery.min.js"></script>
<script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquui/1.8.18/jquery-ui.min.js"></script>
<!−− DataTables CSS −−>
<link rel="stylesheet" type="text/css" href="http://ajax.aspnetcdn.com/ajax/jquery.dataTables/1.8.2/css/jquery.dataTables.css" />
<!−− DataTables −−>
<script type="text/javascript" charset="utf8" src="http://ajax.aspnetcdn.com/ajax/jquery.dataTables/1.8.2/jquery.dataTables.min.js"></script>
<?php
ini_set('display_errors',1);
error_reporting(E_ALL);
echo '</head';
}
function printHeader()
{
    ?>
    <div id="header">
        <h1><a href="home.php" img src="omrs-header-248x72.png"/></a>
        Metadata Sharing Server</h1>
    </div>
<?php
}
function printFooter()
{
    ?>
    <h3>Thanks for using <em>Metadata Sharing Server</em>!</h3><br />
    Page by JWld. 2012.<p>
    <?php
}

// Prints navigation option links
// Takes as parameter the current user for determining user class based options
// Outputs HTML
function printNavOptions($user)
{
    if ($user['class'] == 1)
    {
        ?><h3>You are in the admin user class! </h3>
        <div id="options">
            <a href="managePackages.php">Manage Packages</a><br />
            <a href="manageUsers.php">Manage Users</a><br />
        </div>
    } else
    {
        echo '<div id="options">
    }
        ?><a href="upload.php">Upload</a><br />
        <a href="home.php">Home</a><br />
        <a href="profilepage.php">Profile</a><br />
        <a href="logout.php">Logout</a><br />
        <a href="loginPage.php">Login</a> to a different account<br />
        <a href="newaccount.php">Register</a> a new account<br />
    </div>
<?php
}

// Generates tabs and uses JQuery UI to style them
// Takes as parameters a user
// Beyond the initial query, delegates the actual content of the tabs to calls tabcontent
function tabs($who = false)
{
    // script
    $(document).ready(function() {
        $('#tabs').tabs();
    });
    $(document).ready(function() {
        $('#tab1table').dataTable({
            'bProcessing': true,
            'bServerSide': true,
            'sAjaxSource': 'proc.php',
            'aaSorting': [[ 3, 'desc' ]],
            'bJQueryUI': true,
            'sPaginationType': "full_numbers"
        });
    });
    $(document).ready(function() {
        $('#tab2table').dataTable({
            'bProcessing': true,
            'bServerSide': true,
            'sAjaxSource': 'proc.php',
            'aaSorting': [[ 6, 'desc' ]],
            'bJQueryUI': true,
            'sPaginationType': "full_numbers"
        });
    });
    $(document).ready(function() {
        $('#tab3table').dataTable({
            'bProcessing': true,
            'bServerSide': true,
        });
    });
}
"sAjaxSource": "proc.php",
"aaSorting": [[ 0, "asc" ]],
"bJQueryUI": true,
"sPaginationType": "full_numbers"
} ;

$(document).ready(function() {
$("#tab4table").dataTable( {
 "bProcessing": true,
 "bServerSide": true,
 "sAjaxSource": "proc.php",
 "aaSorting": [[ 2, "asc" ]],
 "bJQueryUI": true,
 "sPaginationType": "full_numbers"
} ) ;

$(document).ready(function() {
$("#tab5table").dataTable( {
 "bProcessing": true,
 "bServerSide": true,
 "sAjaxSource": "proc.php",
 "aaSorting": [[ 5, "desc" ]],
 "bJQueryUI": true,
 "sPaginationType": "full_numbers"
} ) ;

</script>

<div id="tabs">
<ul>
<li><a href="#firsttab">Newest</a></li>
<li><a href="#secondtab">Most Popular</a></li>
<li><a href="#thirdtab">Alphabetical</a></li>
<?php
</ul>

<?php if (!$who) {
</ul>
<br />
<li><a href="#fourthtab">By Author</a></li>
</ul>

<?php
<br />
</ul>

</div id="firsttab">
<table id="tab1table" class="tabtable" border = "5px">
<thead>
<tr><th>Name</th><th>Description</th><th>Authority</th><th>Date Uploaded</th><th>Version</th><th>OpenMRS Version</th><th>Downloads</th></tr>
</thead>
</table>

</div>

</div id="secondtab">
<table id="tab2table" class="tabtable" border = "5px">
<thead>
<tr><th>Name</th><th>Description</th><th>Authority</th><th>Date Uploaded</th><th>Version</th></tr>
</thead>
</table>

</div>
<?php

// Takes as parameters a referring page and a database query
// Displays the data resulting from making the query in the way appropriate
// for the referring page
// Outputs HTML
function tabcontent($query, $off)
{
    ?><script type="text/javascript">
        function confirmDelete(name){
            
            </script>
var confirmed = confirm("Are you sure? This will remove package \\
" + name + \\
" forever.");
return confirmed;

</script>

<?php

/*

echo ' <td><a href="download.php?id=' . $row['id'] . \\
"DL</a>&nbsp';
if($user && ($user['class'] == 1 || ($user['id'] == $row['authid'])))
{
  echo ' <a href="managePackages.php?id=' . $row['id'] . \\
">DL</a>&nbsp';
  if($user['class'] == 1)
  {
    echo ' &ref=maned';
  }
  else
  {
    echo ' &ref=prof';
  }
}
else
{
  echo ' &ref=prof';
}
echo ' <a href="deletePackage.php?id=' . $row['id'] . \\
"ED</a>&nbsp';
echo ' <a href="viewPackages.php?id=' . $row['id'] . \\
">ED</a>&nbsp';
echo ' <a href="deletePackage.php?id=' . $row['id'] . \\
">RM</a>&nbsp';
if($user['class'] == 1)
{
  echo ' &ref=man';
}
else
{
  echo ' &ref=prof';
}
echo " onclick="return confirmDelete(\" . $row['name'] \\
" on click="return confirmDelete(\"");
}
}
dbclose($conn);

else
{
  echo 'No records found for that query.';
}
</tbody>
</table><br />
<?php */


<?php
// General utility functions
// Open a zip archive
// Takes as parameter a path to the archive
// Outputs the extracted archive to an unzip folder in document root
// Returns a boolean success indicator
function openZip($path)
{
    $zip = new ZipArchive;
    if($zip->open($path) == true) {
        for($i = 0; $i < $zip->numFiles; $i++) {
            $filename = $zip->getNameIndex($i);
            $zip->extractTo(dirname(__FILE__) . '/unzip/');
        }
        $zip->close();
        return true;
    } else {
        return false;
    }
}

// Adds a file to a zip archive
// Takes as parameters a path to the archive to add to, a path to the file to add, and the name of the file to add
// Returns a boolean success indicator
function writeZip($archpath, $filepath, $filename)
{
    $zip = new ZipArchive;
    if($zip->open($archpath) == TRUE) {
        $zip->addFile($filepath, $filename);
        $zip->close();
        return true;
    } else {
        return false;
    }
}

// Redirects the browser to a different page
// Takes as parameters the page url to redirect to and a delay in seconds
// Outputs HTML
function redirect($url, $delay = 0)
{
    echo '<META HTTP-EQUIV="REFRESH" CONTENT="' . $delay . '; URL=' . $url . '" >';
}

// Validates that the current visitor to the page is logged in and has a valid session ID
// Returns the validated user or false if none
function getAuthenticatedUser()
{
    $ret = false;
    if(isset($_SESSION['email']) && isset($_SESSION['id'])) {
        
}
// Open the connection with MySQL
$conn = dbconnect();
$email = mysql_real_escape_string($_SESSION['email']);
$query = "SELECT * FROM 'users' WHERE 'email' = '$email';
$result = mysql_query($query);
$isSesh = mysql_num_rows($result);
$person = mysql_fetch_array($result);

dbclose($conn);

if($isSesh == 1)
{
    $ret = $person;
}
return $ret;

// helper function if you don't need user details

// returns the authentication status
function isAuthenticated()
{
    return (getAuthenticatedUser() != false);
}

// Deletes a file and redirects the browser
// Takes as parameters a path to the file to delete and a target url to redirect to
function deleteFile($path, $target = 'home.php')
{
    // File Exists?
    if(file_exists($path))
    {
        unlink($path);
    }
    else
    {
        redirect($target, 5);
        die('File Not Found');
    }

?>
F.4 Functions

dbFuncs.php

```php
<?php

// Database functions
// Connect to the MySQL database,
// Returns the open connection
function dbconnect() {
    // Open the connection with MySQL
    $conn = mysql_connect($GLOBALS['dbhost'], $GLOBALS['dbuser'], $GLOBALS['dbpass']) or die('Error connecting to mysql');
    // Connect to the database in MySql
    mysql_select_db($GLOBALS['dbname']);
    return $conn;
}

// Close a connection to the MySQL database
// Takes as parameter a connection
function dbclose($conn) {
    mysql_close($conn); // Close the connection with MySQL
}

?>
```

pwFuncs.php

```php
<?php

// Password security functions
// Returns a random salt to add to the password to help discourage rainbow table attacks on a stolen db
function getPasswordSalt() {
    return substr(str_pad(dechex(mt_rand()), 8, '0', STR_PAD_LEFT), -8);
}

// Takes as parameters an 8−bit salt and a plaintext password
// Calculates and returns the hash from the salt and the password
function getPasswordHash( $salt, $password ) {
    // TODO: use other hash function if possible SHA1 has known collisions
    return $salt . sha1($salt . $password);
}

// Takes as parameters a plain text password and the stored hash
// Compares the two parameters and returns a boolean
function comparePassword($password, $hash) {
    $salt = substr($hash, 0, 8);
    return $hash == getPasswordHash($salt, $password);
}

?>
```
<?php
include_once 'includes.php';
startPage('Home');
?>
<script type="text/javascript">
  var searchString, type1, type2, data;
  $(function () {
    "use strict";
    $('.search_button').click(function () {
      // getting the value that user typed
      searchString = $('.search_box').val();
      type1 = $('.search_name').is(':checked');
      type2 = $('.search_auth').is(':checked');
      // forming the queryString
      data = 'search=' + searchString + '&type1=' + type1 + '&type2=' + type2;
      // if searchString is not empty
      if (searchString) {
        // ajax call
        $.ajax({
          type: "POST",
          url: "search.php",
          data: data,
          beforeSend: function () { // this happens before actual call
            $('.results').html('');
            $('.searchresults').show();
            $('.word').html(searchString);
          },
          success: function (html) { // this happens after we get results
            $('.results').show();
            $('.results').append(html);
          }
        });
      }
      return false;
    });
  });
</script>
</head>
<body>
<?php printHeader();?>
<h1>Home</h1>
<div id="container">
  <div style="margin:20px auto; text-align: center;">
    <form>
      <input type="text" name="search" id="search_box" class="search_box" />
      <input type="submit" value="Search" class="search_button" />
      <br />
      <input type="radio" name="searchtype" value="1" id="search_name" class="search_type" checked />By Name
      <input type="radio" name="searchtype" value="0" id="search_desc" class="search_type" />By Description
      <input type="radio" name="searchtype" value="2" id="search_auth" class="search_type" />By Author
    </form>
  </div>
</div>
<?php
tabs();

$user = getAuthenticatedUser();
if($user)
{
    echo '<h3>Welcome, ' . $user['name'] . '!</h3>';
    printNavOptions($user);
}
else
{
    <a href="loginPage.php">Login</a> to Upload your favorite metadata package!<br />
    New User? Register <a href="newaccount.php">Here</a><br />
    <?php
    }
    printFooter();
?>
</body>
</html>
F.6 newaccount.php

```php
<?php
include_once 'includes.php';
startPage('New Account');
echo '<body';
printHeader();?
<script type="text/javascript" src="formvalidate.js"></script>
<h2>New User Registration</h2>
<div id="registration">
Please fill out the following fields:<br/>
<form name="new" action="addUser.php" onsubmit='return checkRequired("new")' method="post">
<table>
<tr><td>Name:</td><td> <input type="text" name="name" /></td></tr>
<tr><td>Email:</td><td> <input type="text" name="email" /></td></tr>
<tr><td>Password:</td><td> <input type="password" name="pass" /></td></tr>
</table>
<input type="submit" value="Submit" />
</form>
<br/><br/><br/>
Already registered? Login <a href="loginPage.php">here</a>!<br/>
<a href="home.php">Home</a>
<?php printFooter();?>
</body>
</html>
```
F.7 loginPage.php

```php
<?php

include_once 'includes.php';
startPage('Login Page');

echo '<body'>;
printHeader();
?

<h2>Login:</h2>
<div id="login">

<?php if(isset($_GET['ref']) && $_GET['ref'] == "bad") {
    echo '<p>That username/password combination was invalid.
        Please try again.</p>'; }

    <form name="new" action="login.php" onsubmit='return loginRequired("new")' method="post">
        <table>
            <tr><td>Email:</td> <td> <input type="text" name = "email" /></td> <td> <div id="emailcheck"></div></td></tr>
            <tr><td>Password:</td> <td> <input type="password" name = "pass" /></td> <td> <div id="passcheck"></div></td></tr>
        </table>
        <br />
        <input type="submit" value="Submit" />
    </form>

    New user? Register <a href="newaccount.php">here</a>!
    <br />
    <a href="home.php">Home</a>

    <?php printFooter(); ?>

</div>
<br />
</body>
</html>
```
F.8 profilepage.php

```php
<?php
include_once 'includes.php';
startPage('Your Profile');
echo '<body'>;
printHeader();
$user = getAuthenticatedUser();
if($user) {
// Open the connection with MySQL
$conn = dbconnect();
$auth = $user['name'];
$packages = $user['packages'];
$query = "SELECT * FROM metadata where auth = '$auth';
$table = mysql_query($query);
dbclose($conn);
echo "<h3>Welcome, " . $user['name'] . "! </h3>";
if(isset($_GET['ref']) && $_GET['ref'] == 'up') {
    echo '<h3>Thank You for Uploading a New Package!</h3>';
} else if(isset($_GET['ref']) && $_GET['ref'] == 'rm') {
    echo '<h3>Package Removed Successfully!</h3>';
} else if(isset($_GET['ref']) && $_GET['ref'] == 'ped') {
    echo '<h3>Package Edited Successfully!</h3>';
} else if(isset($_GET['ref']) && $_GET['ref'] == 'ued') {
    echo '<h3>User account details Edited Successfully!</h3>';
}
printNavOptions($user);
?<div id="details">
    Name:&nbsp <?php echo $user['name'];?><br />
    Email Address:&nbsp <?php echo $user['email'];?>
    ?<br />
    Member Since:&nbsp <?php echo $user['joindate'];?>
    ?<br />
    Packages Uploaded:&nbsp <?php echo $user['packages'];?>
    ?<br />
    Edit</a><br />
    <h3>Your Uploaded Packages:</h3>
    <?php
tabs($auth);
    echo '</div>';?
} else // Invalid session
    redirect('loginPage.php');
} printFooter();?
</body></html>
```
F.9  upload.php

```php
<?php
    include_once 'includes.php';
    startPage('Upload Package');
    echo '<body>';
    printHeader();
    $user = getAuthenticatedUser();
    // Display if Logged in
    if ($user)
        { ?>
            <h3>Welcome &nbsp <?php echo $user['name'] ?>!</h3>
            <h3>Upload a Package to Share:</h3>
            <p>Choose a file:</p>
            <form enctype="multipart/form-data" action="uploader.php" onsubmit="" method="POST">
                <input type="hidden" name="MAX_FILE_SIZE" value="1000000" />
                <input name="uploadedPackage" type="file" />
                <input type="submit" value="Upload Package" />
            </form>
        <?php
            printFooter();
        } else
        { ?>
            <h3>You must be logged in to upload a package.</h3>
            <a href="loginPage.php">Login</a>
            <a href="home.php">Home</a>
        <?php
    } ?>
</body></html>
```
F.10 managePackages.php

```php
<?php
include_once 'includes.php';
startPage('Manage Packages');
echo '<body>';
printHeader();
$user = getAuthenticatedUser();
if($user)
{
    if(isset($_POST['ref']) && $_POST['ref'] == 'prof')
    {
        redirect('profilepage.php');
    }
    echo '<h3>Welcome, ' . $user['name'] . '!</h3>
printNavOptions($user);
    if(isset($_GET['ref']) && $_GET['ref'] == 'rm')
    {
        echo '<h3>Package Removed Successfully</h3>';
    }
    else if(isset($_GET['ref']) && $_GET['ref'] == 'ed')
    {
        echo '<h3>Package Edited Successfully</h3>';
    }
    else if(isset($_GET['ref']) && isset($_GET['id']) && ($_GET['ref'] == 'maned' || $_GET['ref'] == 'prof'))
    {
        // Open the connection with MySQL
        $conn = dbconnect();
        $edid = mysql_real_escape_string($_GET['id']);
        $query = "SELECT * FROM `metadata` WHERE `id` = '$edid';";
        $row = mysql_query($query);
        $pack = mysql_fetch_array($row);
        $packExists = mysql_num_rows($row);
        dbclose($conn);
        if($packExists == 1 && ($user['class'] == 1 || $pack['authid'] == $user['id']))
        {
            ?><h3>Editing a Package:</h3>
            <form name="edit" action="peditor.php?ref=<?php echo $_GET['ref']; ?>" method="post">
                <input type="hidden" name="id" value="<?php echo $edid; ?>" />
                <tr><td>Package Name:</td><td><input type="text" name="name" value="<?php echo $pack['name']; ?>" /></td></tr>
                <tr><td>Description:</td><td><input type="text" name="DESC" value="<?php echo $pack['desc']; ?>" /></td></tr>
            </form>
            <p>(note: changes will only take effect if you click save)</p>
        } else if($user['class'] == 1)
        { ... }
    }
?>
</body>
```
```php
51    {  
52        echo '<h3>Here are our uploaded packages</h3>';
53        tabs();
54    }
55
56    else
57    {
58        echo '<p>You must be logged in to manage metadata packages! &nbsp You will be redirected to the login page in a few seconds...</p>';
59        redirect('loginPage.php', 5);
60    }
61    printFooter(); ?>
62    </body></html>
```
F.11 manageUsers.php

```php
<?php
include_once 'includes.php';
startPage('Manage Users');
echo '<body';
printHeader();
$user = getAuthenticatedUser();
if ($user)
    {
    echo "<h3>Welcome, " . $user['name'] . "! </h3>";
    printNavOptions($user);
    if(isset($_GET['ref']) &amp; $_GET['ref'] == 'ed')
    {
    echo '<h3>User Account Details Edited Successfully </h3>' ;
}
    else if(isset($_GET['ref']) &amp; $_GET['ref'] == 'rm')
    {
    echo '<h3>User Account Removed Successfully </h3>' ;
}
    else if(isset($_GET['ref']) &amp; $_GET['ref'] == 'edf')
    {
    echo '<h3>There was an error editing user account details </h3>' ;
}
    else if(isset($_GET['id']) &amp; isset($_GET['ref']) &amp; $_GET['ref'] == 'man')
    {
    // Open the connection with MySQL
    $conn = dbconnect();
    $edid = mysql_real_escape_string($_GET['id']);
    $query = "SELECT * FROM 'users' WHERE 'id' = '$edid'" ;
    $row = mysql_query($query);
    $edu = mysql_fetch_array($row);
    $eduExists = mysql_num_rows($row);
    if($eduExists == 1 &amp;&amp; ($user['class'] == 1))
        {
        $pw = getPasswordHash(getPasswordSalt(), 'MDSSUser-' . $edu['name']);
        $query = 'UPDATE 'users' SET 'pw' = '$pw' WHERE 'id' = '$edid'
        $row = mysql_query($query);
        }
    dbclose($conn);
    echo '<h3>Password reset successfully </h3>' ;
}
else if(isset($_GET['id']) &amp; isset($_GET['ref']) &amp; ($_GET['ref'] == 'maned' || $_GET['ref'] == 'prof'))
// TODO: validate new email changes with javascript
    {
    // Open the connection with MySQL
    $conn = dbconnect();
    $edid = mysql_real_escape_string($_GET['id']);
    $query = "SELECT * FROM 'users' WHERE 'id' = '$edid'" ;
    $row = mysql_query($query);
    $edu = mysql_fetch_array($row);
    $eduExists = mysql_num_rows($row);
    }
```
dbclose($conn);

if($eduExists == 1 && ($user['class'] == 1 || $edu['id'] == $user['id'])) {
  // TODO: javascript to verify
  ?><h3>Editing User&nbsp<?php echo $edu['id']; ?>: </h3>
  <form name="edit" action="ueditor.php?ref=<?php echo $_GET['ref']; ?>" onsubmit=' return checkRequired("new")' method="post">
    <input type="hidden" name="id" value="<?php echo $edid; ?>" />
    <tr><td>Name:</td> <td> <input type="text" name="name" value="<?php echo $edu['name']; ?>" /> </td> <td> <div id="namecheck"></div></td> <br /></tr>
    <tr><td>Email: </td> <td> <input type="text" name="email" value="<?php echo $edu['email']; ?>" /> </td>
  <td> <div id="emailcheck"></div></td> <br /></tr>
<?php if($user['class'] == 0) {
  ?><tr><td>Current Password: </td> <td> <input type="password" name="pass" /> </td>
  <td> <div id="passcheck"></div></td> <br /></tr>
<?php } ?></form>

<?php if($user['class'] == 1) {
  ?><tr><td>User Class: <br /></td> <td> <input type="radio" name="class" value="admin"<?php if($edu['class'] == 1) echo ' checked'; ?>> Admin</td>
  <td> <input type="radio" name="class" value="user"<?php if($edu['class'] == 0) echo ' checked'; ?>> User</td> <br /></tr>
<?php } ?></tr><input type="submit" value="Save Changes" />
  <a href="<?php if($_GET['ref'] == 'maned') {
    echo 'manageUsers';
  } else if($_GET['ref'] == 'prof') {
    echo 'profilepage';
  } ?>.php"><input type="button" name="cancel" value="Cancel" /></a>
  </form>
</p>(note: changes will only take effect if you click save)
</p>
<?php

if($user['class'] == 1 && !isset($_GET['ref']) || ($_GET['ref'] != 'maned' && $_GET['ref'] != 'prof'))
{
  ?><script type="text/javascript">
    function confirmDelete(name){
      var confirmed = confirm("Are you sure? This will remove user " + name + ": and all their packages forever.");
      return confirmed;
    }
  </script>
  Here are our registered users:<br />
  <table border="5px">
    <tr><td> name </td><td> email </td><td> class </td><td> join date </td><td> packages </td><td> Actions </td></tr>
<?php
  // Open the connection with MySQL
  $conn = dbconnect();
  $query = "SELECT * FROM 'users'";
  $table = mysql_query($query);
  while($row = mysql_fetch_array($table))
  {
    echo '<td>' . $row['name'] . '</td>';
    echo '<td>' . $row['email'] . '</td>';
    if($row['class'] == 1)
    {
      echo '<td>Admin</td>';
    }
    else
    {
      echo '<td>User</td>';
    }
    echo '<td>' . $row['joindate'] . '</td>';
    echo '<td>' . $row['packages'] . '</td>';
    echo '<td><a href="manageUsers.php?id=' . $row['id'] . '&ref=man">Reset PW</a>&nbsp' . '<a href="manageUsers.php?id=' . $row['id'] . '&ref=maned">Edit</a>&nbsp' . '<a href="removeUser.php?id=' . $row['id'] . '&ref=manrm" onclick="return confirmDelete("" . $row['name'] . ' ")">Remove</a></td>
    </tr>
  }
  dbclose($conn);
  echo '</table>';
}</script>
else
{
  // No such user or corrupt session
  redirect('loginPage.php');
}?>
</body></html>
F.12 search.php

```php
<?php
// if we got something through $_POST
if (isset($_POST['search']) && isset($_POST['type1']) && isset($_POST['type2'])) {
    include_once 'includes.php';
    // Open the connection with MySQL
    $conn = dbconnect();
    $word = '%'. mysql_real_escape_string($_POST[ 'search' ]). '%';
    dbclose($conn);
    if ($_POST[ 'type1' ] == 'true') {
        $query = "SELECT * FROM 'metadata' WHERE 'name' LIKE '$word' ORDER BY 'dls' DESC LIMIT 10";
    } else if ($_POST[ 'type2' ] == 'true') {
        $query = "SELECT * FROM 'metadata' WHERE 'auth' LIKE '$word' ORDER BY 'dls'";
    } else {
        $query = "SELECT * FROM 'metadata' WHERE 'desc' LIKE '$word' ORDER BY 'dls' DESC LIMIT 10";
    }
    tabcontent('main', $query);
    /* code for bolding the search term in a result */
    if ($isEntry >= 1 && isset($row)) {
        $end_result = '<td >';
        foreach ($row as $r) {
            $res = $r[ 'name' ];
            // bold the search word in result
            $bold = '<span class="found">'. $word . '</span>'; $word . '</td >';
            $end_result .= '<tr >'. str_replace($word, $bold, $res) . '</tr >';
        }
        $end_result .= '</td >';
        echo $end_result;
    } else {
        echo '<li>No results found</li >';
    }
*/
?>
```
F.13 download.php

```php
<?php
include_once 'includes.php';
startPage('Download');

function downloadFile($fullPath, $filename)
{
    // Must be fresh start
    if(headers_sent())
    {
        die('headers Sent');
    }
    // Required for some browsers
    if(ini_get('zlib.output_compression'))
    {
        ini_set('zlib.output_compression', 'Off');
    }
    // File Exists?
    if(file_exists($fullPath))
    {
        // Parse Info / Get Extension
        $fsize = filesize($fullPath);
        $ctype="application/x-zip-compressed";
        header("Pragma: public"); // required
        header("Expires: 0");
        header("Server: ");
        header("X-Powered-By: ");
        header("Cache-Control: must-revalidate, post-check=0, pre-check=0");
        header("Cache-Control: private", false); // required for certain browsers
        header("Content-Type: ". $ctype);
        header("Content-Disposition: attachment; filename=" . $filename . ";");
        header("Content-Transfer-Encoding: binary");
        header("Content-Length: ". $fsize);
        ob_clean();
        flush();
        readfile($fullPath);
    }
    else
    {
        die('File Not Found');
    }
}

if(isset($_GET['id']))
{
    // Open the connection with MySQL
    $conn = dbconnect();
    $id = mysql_real_escape_string($_GET['id']);
    $query = "SELECT * FROM 'metadata' WHERE 'id' = '$id';"
    $table = mysql_query($query);
    $isEntry = mysql_num_rows($table);
    $row = mysql_fetch_array($table);
    $dls = $row['dls'] + 1;
    if($isEntry == 1)
```
{ downloadFile($row["path"], $row["name"]) ;
$query = "UPDATE 'metadata' SET 'dls' = '$dls' WHERE 'id' = '$id'" ;
$result = mysql_query($query) ;
}
dbclose($conn) ;
?>
</body></html>
F.14 addUser.php

```php
<?php
include_once 'includes.php';
startPage('User Registration');
echo '<body>';
printHeader();
if(isset($_POST['name']) && isset($_POST['email']) && isset($_POST['pass'])) {
    // Open the connection with MySQL
    $conn = dbconnect();
    $name = trim(mysql_real_escape_string($_POST['name']));
    $email = trim(mysql_real_escape_string($_POST['email']));
    $pass = trim($_POST['pass']);
    if($email == '') {
        echo '<p>Please enter an email address! &nbsp You will be redirected to the previous page in a few seconds...</p>';
        redirect('newaccount.php', 5);
    } else if($pass == '') {
        echo '<p>Please enter a password! &nbsp You will be redirected to the previous page in a few seconds...</p>';
        redirect('newaccount.php', 5);
    } else if($name == '') {
        echo '<p>Please enter a username! &nbsp You will be redirected to the previous page in a few seconds...</p>';
        redirect('newaccount.php', 5);
    } else {
        // Check if the user already exists in the table
        $query = "SELECT * FROM 'users' WHERE 'email' = '$email'";
        $result_user_check = mysql_query($query);
        $isEntry = mysql_num_rows($result_user_check);
        // Add user if not already in the database
        if($isEntry == 0) {
            // get a new hash for a password
            $hash = getPasswordHash(getPasswordSalt(), $pass);
            $defaultPackages = 0;
            $joinDate = date(time());
            $query = "INSERT INTO 'users' ('name', 'email', 'pw', 'class', 'joindate', 'packages') VALUES ('$name', '$email', '$hash', '0', FROM_UNIXTIME('$joinDate'), '$defaultPackages')";
            mysql_query($query);
            echo '<p>New User Added! Try to log in now &lt;a href="loginPage.php">here</a&gt; &lt;/p&gt;';
        }
    }
} else {
    ?>
```
A user account with that email address already exists. Please log-in <a href="loginPage.php">here</a>.

If you forgot your password click <a href="pwRecovery.php">here</a> to recover.

If you would like to create a new user account please choose a different email <a href="newaccount.php">here</a>.

<?php

dbclose($conn);

else
{
    redirect('home.php');
}
printFooter();
?>
</body>
</html>
F.15 login.php

```php
<?php

Session_start();
include_once 'includes.php';
$val = isAuthenticated();
if($val) // Someone is already logged in – log them out
{
    // Unset all of the session variables.
    $_SESSION = array();
    Session_destroy();
}
ob_start();
startPage('Login');
ob_end_clean();
echo '<body>' ;

// Open the connection with MySQL
$conn = dbconnect();
$email = mysql_real_escape_string($_POST['email']);
$pass = $_POST['pass'];
$query = "SELECT * FROM 'users' WHERE 'email' = '$email';"
$hash = mysql_query($query);
$isEntry = mysql_num_rows($hash);
$row = mysql_fetch_array($hash);
if((($isEntry != 1) || !comparePassword($pass, $row['pw'])))
{
    redirect('loginPage.php?ref=bad');
}
else
{
    $seshid = getPasswordHash(getPasswordSalt(), date(time()));
    $_SESSION['id'] = $seshid;
    $_SESSION['name'] = $row['name'];
    $_SESSION['email'] = $email;
    redirect('profilepage.php');
}
$dbclose($conn);
?>
</body>
</html>
```
F.16 uploader.php

```php
<?php
include_once 'includes.php';
startPage('Uploader');

$person = getAuthenticatedUser();

if($person)
{
    $okay = false;
    $accepted_types = array('application/zip', 'application/x-zip-compressed', 'multipart/x-zip', 'application/x-compressed');

    if(isset($_FILES))
    {
        $file = $_FILES['uploadedPackage'];
    }
    else
    {
        redirect('home.php');
        die("No file.");
    }

    $type = $file['type'];
    foreach($accepted_types as $mime_type) {
        if($mime_type == $type) {
            $okay = true;
            break;
        }
    }

    if($okay && ($file['size'] < 1000000))
    {
        if($file['error'] > 0)
        {
            echo "Error: " . $file['error'] . "<br />";
        }
        else
        {
            echo "Upload: " . $file['name'] . "<br />";
            echo "Type: " . $file['type'] . "<br />";
            echo "Size: " . ($file['size'] / 1024) . " Kb<br />";

            $blacklist = array('.php', '.phtml', '.php3', '.php4');
            foreach ($blacklist as $item)
            {
                if(preg_match("/\$/i", $file['name']))
                {
                    echo "We do not allow uploading PHP files\\n";
                    exit;
                }
            }

            // add new generated filename to path
            $target_path = str_replace('\', '/', $GLOBALS['uploadpath']) . getPasswordHash(getPasswordSalt(), $file['name']) . ".zip";

            if(move_uploaded_file($file['tmp_name'], $target_path))
            {
                echo "The file " . basename($file['name']) . " has been uploaded";
            }
        }
    }
```

// Extract the package info from the header.xml file
// TODO: handle gracefully if this fails
openZip($target_path);

// open the metadata.xml file
// we don’t really care about the contents
// just check that it isn’t corrupted.
$xml = dirname(__FILE__) . '/unzip/metadata.xml';
ob_start();
$x = simplexml_load_file($xml);
ob_end_clean();

if($x == FALSE || empty($x))
{
    deleteFile($target_path);
    redirect('home.php', 5);
    die('<br /><h3> WARNING! Bad xml file. Please check your
    file and try again.</h3>');
}

// now open the header.xml file which contains the info we
// want
$xml = dirname(__FILE__) . '/unzip/header.xml';
ob_start();
$x = simplexml_load_file($xml);
ob_end_clean();

if($x == FALSE || empty($x))
{
    deleteFile($target_path);
    redirect('home.php', 5);
    die('<br /><h3> WARNING! Bad xml file. Please check your
    file and try again.</h3>');
}
else
{
    // Open the connection with MySQL
    $conn = dbconnect();
    // $uuid = $x->attributes()->uuid;
    $pubdate = mysql_real_escape_string($x->dateCreated);
    $name = mysql_real_escape_string($x->name);
    $desc = mysql_real_escape_string($x->description);
    $omrsver = mysql_real_escape_string($x->openmrsVersion);
    $ver = mysql_real_escape_string($x->version);
    // $group = mysql_real_escape_string($x->group);
    $suburl = mysql_real_escape_string($x->subscriptionUrl);
    // TODO: capture MDS version here
    $auth = mysql_real_escape_string($person['name']);
    $authid = $person['id'];
    $update = date(time());
    $query = "SELECT * from 'metadata' WHERE 'name' = '$name'
    AND 'omrsver' = '$omrsver' AND 'ver' = '$ver';";
    $row = mysql_query($query);
    $isEntry = mysql_num_rows($row);
    if($isEntry >= 1)
    {
        deleteFile($target_path);
        redirect('upload.php', 5);
        die("<h3>We’re sorry that package has already been
        shared. Please try a different package.</h3>");
    }
}
$query = "INSERT INTO 'metadata' ('name', 'path', 'auth', 'authid', 'pubdate', 'update', 'desc', 'omrsver', 'ver', 'suburl', 'dls')";
$query .= "VALUES ('$name', '$target_path', '$auth', '$authid', '$pubdate', FROM_UNIXTIME('$update'), '$desc', '$omrsver', '$ver', '$suburl', '0')";
mysql_query($query);

$query = "UPDATE 'users' SET 'packages' = 'packages'+1 WHERE 'id' = '$authid'";
mysql_query($query);
dbclose($conn);

redirect('profilePage.php?ref=up', 5);
else
{
    echo '<p>There was an error uploading the file, please try again!</p>';
    redirect('home.php', 5);
}
else
{
    echo '<p>Invalid file. Please try again.</p>';
    redirect('home.php', 5);
}?>
</body>
</html>
<?php
ini_set('display_errors',1);
error_reporting(E_ALL);
Session_start();
include_once 'includes.php';
$user = getAuthenticatedUser();
// if Logged in
if($user)
{
    if(isset($_POST['id']))
    {
        // Open the connection with MySQL
        $conn = dbconnect();
        $pid = mysql_real_escape_string($_POST['id']);
        $query = "SELECT * FROM 'metadata' WHERE 'id' = '$pid';";
        $row = mysql_query($query);
        $pack = mysql_fetch_array($row);
        $packExists = mysql_num_rows($row);
        // close the connection with MySQL
        dbclose($conn);
        if($packExists == 1 && ($user['class'] == 1 || $pack['auth'] == $user['name'])
        {
            if(isset($_POST['name']) && isset($_POST['desc']))
            {
                // Open the connection with MySQL
                $conn = dbconnect();
                $name = trim(mysql_real_escape_string($_POST['name']));
                $desc = trim(mysql_real_escape_string($_POST['desc']));
                $ver = $pack['ver'];
                $omrsver = $pack['omrsver'];
                $query = "SELECT * from 'metadata' WHERE 'name' = '$name'
                            AND 'ver' = '$ver' AND 'omrsver' = '$omrsver';";
                $row = mysql_query($query);
                $isEntry = mysql_num_rows($row);
                if($isEntry >= 1 && $pack['name'] != $name)
                {
                    if(isset($_GET['ref'])
                    {
                        if($_GET['ref'] == 'prof')
                        {
                            redirect('profilepage.php', 5);
                        }
                    } else if($_GET['ref'] == 'maned')
                    {
                        redirect('managePackages.php', 5);
                    }
                    else
                    {
                        redirect('home.php', 5);
                    }
                } else
                } else
                {
```php
    die("We're sorry a package with that name has already been shared. Please try a different name.");

    $query = "UPDATE 'metadata' SET 'name' = '{$name}', 'desc' = '{$desc}' WHERE 'id' = '{$pid}';
    mysql_query($query);
    // close the connection with MySQL
dbclose($conn);
    $target_path = $pack['path'];
    openZip($target_path);
    $xml = dirname(__FILE__) . '/unzip/header.xml';
    $x = simplexml_load_file($xml);
    // TODO: decide which other fields to edit here
    $x->name = $name;
    $x->description = $desc;
    if(file_put_contents($xml, $x->asXML()) !== false)
    {
        redirect('managePackages.php', 5);
        die('Failed to edit xml.');
    }
    writeZip($target_path, $xml, 'header.xml');

    if(isset($_GET['ref']))
    {
        if($_GET['ref'] == 'prof')
        {
            redirect('profilepage.php?ref=ped');
        }
        else if($_GET['ref'] == 'maned')
        {
            redirect('managePackages.php?ref=ed');
        }
    }
    else
    {
        redirect('home.php');
    }    ```
<?php
ini_set('display_errors', 1);
error_reporting(E_ALL);
Session_start();
include_once 'includes.php';
$user = getAuthenticatedUser();
if($user) // if Logged in
{
    if(isset($_POST['id']))
    {
        // Open the connection with MySQL
        $conn = dbconnect();
        $uid = mysql_real_escape_string($_POST['id']);
        $query = "SELECT * FROM 'users' WHERE 'id' = '$uid'";
        $row = mysql_query($query);
        $edu = mysql_fetch_array($row);
        $uExists = mysql_num_rows($row);
        if($uExists == 1 && ($user['class'] == 1 || $edu['id'] == $user['id']))
        {
            if(isset($_POST['name']) && isset($_POST['email']) && isset($_POST['newpw']) && isset($_POST['newpw2']))
            {
                if($_POST['newpw'] != $_POST['newpw2'])
                {
                    if(isset($_GET['ref']) )
                    {
                        if($_GET['ref'] == 'prof')
                        {
                            redirect('profilepage.php', 5);
                        }
                        else if($_GET['ref'] == 'maned')
                        {
                            redirect('manageUsers.php', 5);
                        }
                    }
                    else
                    {
                        redirect('home.php', 5);
                    }
                    die("All passwords must match. Please check your
typing and try again.");
                }
            }else
            {
                $name = trim(mysql_real_escape_string($_POST['name']));
                $email = trim(mysql_real_escape_string($_POST['email']));
                $query = "SELECT * from 'users' WHERE 'name' = '$name'";
                $row = mysql_query($query);
                $isEntryN = mysql_num_rows($row);
            }
            $query = "SELECT * from 'users' WHERE 'email' = '$email'";
            $row = mysql_query($query);
            $isEntryE = mysql_num_rows($row);
        }else
        {
            redirect('home.php', 5);
        }
if(($isEntryN >= 1 && ($name != $edu['name'])) || ($isEntryE >= 1 && ($email != $edu['email'])))
{
    if(isset($_GET['ref']))
    {
        if($_GET['ref'] == 'prof')
        {
            redirect('profilepage.php', 5);
        } else if($_GET['ref'] == 'maned')
        {
            redirect('manageUsers.php', 5);
        }
    } else
    {
        redirect('home.php', 5);
    }
    die("We're sorry a user with that name or email already exists. Please try a different name or email.");
} else
{
    $authid = $edu['id'];
    $query = "UPDATE 'metadata' SET 'auth' = '$name' WHERE 'authid' = '$authid'";
    mysql_query($query);
    $query = "UPDATE 'users' SET 'name' = '$name', 'email' = '$email' WHERE 'id' = '$uid'";
    mysql_query($query);
    if($_POST['newpw'] != '')
    {
        if($_POST['class'] == 1 || isset($_POST['pass']) && (comparePassword(trim($_POST['pass']), $_POST['pw']) == TRUE))
        {
            $newpw = getPasswordHash(getPasswordSalt(), trim($_POST['newpw']));
            $query = "UPDATE 'users' SET 'pw' = '$newpw' WHERE 'id' = '$uid'";
            mysql_query($query);
        } else
        {
            if(isset($_GET['ref']))
            {
                if($_GET['ref'] == 'prof')
                {
                    redirect('profilepage.php', 5);
                } else if($_GET['ref'] == 'maned')
                {
                    redirect('manageUsers.php', 5);
                }
            } else
            {
                // Code...
            }
        }
    }
}
redirect('home.php', 5);
}

die("All passwords must match. Please check your typing and try again.");

if($user['class'] == 1 && isset($_POST['class']))
{
    $class = ($_POST['class'] == 'admin') ? 1 : 0;
    $query = "UPDATE 'users' SET 'class' = 'class'
    WHERE 'id' = '$uid';
    mysql_query($query);
}

// close connection to MySQL
$dbclose($conn);

if(isset($_GET['ref']) && $_GET['ref'] == 'maned')
{
    redirect('manageUsers.php?ref=ed');
}

// TODO: indicate failures here
else if(isset($_GET['ref']) && $_GET['ref'] == 'maned' && false)
{
    redirect('manageUsers.php?ref=edf');
}
else if(isset($_GET['ref']) && $_GET['ref'] == 'prof')
{
    redirect('profilepage.php?ref=ued');
}
else
{
    redirect('home.php');
}

else
{
    redirect('home.php');
}
F.19 deletePackage.php

```php
<?php
include_once 'includes.php';
startPage('Deleting Package...');
echo '<body>';

if(isset($_GET['id']))
{
    $person = getAuthenticatedUser();
    if($person)
    {
        // Open the connection with MySQL
        $conn = dbconnect();
        $id = mysql_real_escape_string($_GET['id']);
        $query = "SELECT * FROM 'metadata' WHERE 'id' = '$id';
        $table = mysql_query($query);
        $isEntry = mysql_num_rows($table);
        $row = mysql_fetch_array($table);
        dbclose($conn);
        if($isEntry == 1 && ($row['auth'] == $person['name'] || $person['class'] == 1))
        {
            if(isset($_GET['ref']) && $_GET['ref'] == 'prof')
            {
                deleteFile($row['path'], 'profilePage.php');
            }
            else if(isset($_GET['ref']) && $_GET['ref'] == 'man')
            {
                deleteFile($row['path'], 'managePackages.php');
            }
            else
            {
                deleteFile($row['path']);
            }

            $conn = dbconnect();
            $query = "DELETE FROM 'metadata' WHERE 'id' = '$id';
            $result = mysql_query($query);
            $id = $row['authid'];
            $query = "UPDATE 'users' SET 'packages' = 'packages'-1 WHERE 'id' = '$id';
            $result = mysql_query($query);
            dbclose($conn);
        }
    }

    if(isset($_GET['ref']) && $_GET['ref'] == 'prof')
    {
        redirect('profilePage.php?ref=rm');
    }
    else if(isset($_GET['ref']) && $_GET['ref'] == 'man')
    {
        redirect('managePackages.php?ref=rm');
    }
    else
    {
        redirect('home.php');
    }
    echo '</body>';
}
```

Princeton University
else
{
    redirect('home.php');
}

</body></html>
<?php
    include_once 'includes.php';
    startPage('Remove User');
    echo '<body>

    $person = getAuthenticatedUser();
    if(isset($_GET['id']) && $person)
    {
        if($person['class'] == 1)
        {
            // Open the connection with MySQL
            $conn = dbconnect();
            $id = mysql_real_escape_string($_GET['id']);
            $query = "SELECT * FROM 'users' WHERE 'id' = '$id';
            $tmp = mysql_query($query);
            $isAuth = mysql_num_rows($tmp);
            $user = mysql_fetch_array($tmp);
            $auth = $user['name'];
            if($isAuth == 1)
            {
                $query = "SELECT * FROM 'metadata' WHERE 'auth' = '$auth';
                $table = mysql_query($query);
                $isEntry = mysql_num_rows($table);
                if($isEntry >= 1)
                {
                    while($row = mysql_fetch_array($table))
                    {
                        deleteFile($row['path']);
                        $pid = $row['id'];
                        $query = "DELETE FROM 'metadata' WHERE 'id' = '$pid';
                        $result = mysql_query($query);
                    }
                }
                $query = "DELETE FROM 'users' WHERE 'id' = '$id';
                $result = mysql_query($query);
                dbclose($conn);
            }
            if(isset($_GET['ref']) && $_GET['ref'] == 'manrm')
            {
                redirect('manageUsers.php?ref=rm');
            }
            else
            {
                redirect('home.php');
            }
        }
    }
</body></html>
### F.21 Exit pages

**logout.php**

```php
<?php

include_once 'includes.php';
startPage('Logging Out...');

echo '<body';
printHeader();
$val = isAuthenticated();
if($val)
{
    // Unset all of the session variables.
    $_SESSION = array();
    session_destroy();
}
?>
<div id="banner">
    <p>You have successfully logged out. &nbsp You will be redirected to the main page in a few seconds... </p>
</div>
<?php
printFooter();
redirect('home.php', 3); ?>
</body></html>
```

**pwRecovery.php**

```php
<?php

startPage('Password Recovery');

echo '<body';
printHeader();?>

<div id="registration">
    <p>Metadata Sharing Server does not offer password recovery services at this time! Please email an admin to reset your password.</p>
    <form name="new" action='' method="post">
        <table>
            <tr><td>Email:</td><td> <input type="text" name="email" /></td></tr>
        </table>
        <input type="submit" value="Submit" />
    </form>
    <table>
        <tr><td>Email:</td><td><input type="text" name="email" /></td></tr>
    </table>
    <input type="submit" value="Submit" />
</div>

<?php
$user = getAuthenticatedUser();
if($user)
{
    printNavOptions($user);
}
else
{
    Login <a href="loginPage.php">here</a>! <br />
    <a href="home.php">Home</a>
    <?php }
printFooter(); ?>
</body></html>
```
<?php

/* *************** *************** *************** *************** */
/* *************** *************** *************** *************** */
ob_start();
include_once 'includes.php';
ob_end_clean();

/* Array of database columns which should be read and sent back to DataTables. Use a space where you want to insert a non-database field (for example a counter or static image) */
$aColumns = array( 'name', 'desc', 'auth', 'update', 'ver', 'omrsver', 'dls' );
/* Indexed column (used for fast and accurate table cardinality) */
$sIndexColumn = "id";
/* DB table to use */
$sTable = $GLOBALS['dttable'];

/* Database connection information */
$gaSql['user'] = $GLOBALS['dbuser'];
$gaSql['password'] = $GLOBALS['dbpass'];
$gaSql['db'] = $GLOBALS['dbname'];
$gaSql['server'] = $GLOBALS['dbhost'];

/* *************** *************** *************** *************** */
/* *************** *************** *************** *************** */
/* If you just want to use the basic configuration for DataTables with PHP server-side, there is no need to edit below this line */
/* */
/* MySQL connection */
/* */
$gaSql['link'] = mysql_pconnect ( $gaSql['server'], $gaSql['user'], $gaSql['password'] ) or die( 'Could not open connection to server' );
mysql_select_db( $gaSql['db'], $gaSql['link'] ) or die( 'Could not select database'. $gaSql['db'] );

/* */
/* Paging */
/* */
$sLimit = "";
if ( isset( $_GET['iDisplayStart'] ) &amp; $_GET['iDisplayLength'] != '-l' ) {
  $sLimit = "LIMIT ",mysql_real_escape_string( $_GET['iDisplayStart'] ) ,". ", ". ",mysql_real_escape_string( $_GET['iDisplayLength'] ) ;
}
/* */
/* Ordering */
/* */
$sOrder = "";
if ( isset($_GET['iSortCol_0']) ) {  
    $sOrder = "ORDER BY ";  
    for ( $i=0 ; $i<intval($_GET['iSortingCols']) ; $i++ ) {  
        if ( $_GET['bSortable_'.$iSortCol_.$i] == "true" ) {  
            $sOrder .= "'" . $aColumns[intval($_GET['iSortCol_'.$i])] . "' LIKE '%" . mysql_real_escape_string($_GET['sSortDir_'.$i]) . ";", ";";  
        }  
    }  
    $sOrder = substr_replace($sOrder, ";", -2);  
    if ( $sOrder == "ORDER BY" ) {  
        $sOrder = "";  
    }  
}  

/**  
 * Filtering  
 * NOTE this does not match the built-in DataTables filtering which  
 * does it word by word on any field. It's possible to do here, but concerned  
 * about efficiency  
 * on very large tables, and MySQL's regex functionality is very  
 * limited  
 */  
$sWhere = ";";  
if ( isset($_GET['sSearch']) && $_GET['sSearch'] != "" ) {  
    $sWhere = "WHERE (";  
    for ( $i=0 ; $i<count($aColumns) ; $i++ ) {  
        $sWhere .= "'" . $aColumns[$i] . "' LIKE '%" . mysql_real_escape_string($_GET['sSearch_' . $i]) . ";", ";";  
    }  
    $sWhere = substr_replace($sWhere, ";", -3);  
    $sWhere .= ");";  
}  

/** Individual column filtering */  
for ( $i=0 ; $i<count($aColumns) ; $i++ ) {  
    if ( isset($_GET['bSearchable_'.$i]) && $_GET['bSearchable_'.$i] == "true" && $_GET['sSearch_'.$i] != "" ) {  
        if ( $sWhere == "" ) {  
            $sWhere = "WHERE ";  
        }  
        else  
        {  
            $sWhere .= " AND ";  
        }  
        $sWhere .= "'" . $aColumns[$i] . "' LIKE '%" . mysql_real_escape_string($_GET['sSearch_'.$i]) . ";", ";";  
}
";

}  /* SQL queries */

/* Get data to display */

$sQuery = "SELECT SQL_CALC_FOUND_ROWS \n\n" . str_replace(" , ", " , implode(" , ", $aColumns))." ;

FROM $sTable

$sWhere
$sOrder
$sLimit

$rResult = mysql_query( $sQuery , $gaSql['link'] ) or die(mysql_error());

/ * Data set length after filtering */

$sQuery = "SELECT FOUND_ROWS()"

$rResultFilterTotal = mysql_query( $sQuery , $gaSql['link'] ) or die(mysql_error());

$aResultFilterTotal = mysql_fetch_array($rResultFilterTotal);

$iFilteredTotal = $aResultFilterTotal[0];

/ * Total data set length */

$sQuery = "SELECT COUNT('".$sIndexColumn."')"

FROM $sTable

$rResultTotal = mysql_query( $sQuery , $gaSql['link'] ) or die(mysql_error());

$aResultTotal = mysql_fetch_array($rResultTotal);

$iTotal = $aResultTotal[0];

/ * Output */

$output = array("sEcho" => intval($_GET['sEcho'])),

"iTotalRecords" => $iTotal,

"iTotalDisplayRecords" => $iFilteredTotal,

"aaData" => array()

);

while( $aRow = mysql_fetch_array( $rResult ) )
{
    $row = array();
    for ( $i=0 ; $i<count($aColumns) ; $i++ )
    {
        if ( $aColumns[$i] == "version" )
        {
            /* Special output formatting for 'version' column */
            $row[ ] = ($aRow[$aColumns[$i]] == "0") ? '-' : $aRow[$aColumns[$i] ];
        }
        else if ( $aColumns[$i] != ' ' )
        {
            
        }

    }

}

90  Princeton University
/* General output */
$row[] = $aRow | $aColumns | $i |

} }
} 
$output['aaData'][] = $row;

} echo json_encode( $output ); ?>
F.23 Javascript

formvalidate.js

function isBlank(x) {
    "use strict";
    return (x === null || x === "");
}

function checkRequired(form) {
    "use strict";
    var emailAlert = false, emailblankAlert = false, nameblankAlert = false;
    var passblankAlert = false, name = document.forms[form][0];
    var email = document.forms[form][1], pass = document.forms[form][2];
    var atpos = email.value.indexOf("@");
    var dotpos = email.value.lastIndexOf(".");
    var emailcheck = document.getElementById("emailcheck");
    var namecheck = document.getElementById("namecheck");
    var passcheck = document.getElementById("passcheck");
    if (isBlank(email.value)) {
        email.style.border = '1px Solid Red';
        emailblankAlert = true;
        emailcheck.innerHTML = '<img src="required.gif" alt="Required!" />'; 
    } else {
        if (atpos < 1 || dotpos < atpos + 2 || dotpos + 2 >= email.value.length) {
            email.style.border = '1px Solid Red';
            emailAlert = true;
            emailcheck.innerHTML = '';
        } else {
            email.style.border = '1px Solid Gray';
            emailcheck.innerHTML = '';
        }
    }
    if (isBlank(name.value)) {
        name.style.border = '1px Solid Red';
        nameblankAlert = true;
        namecheck.innerHTML = '<img src="required.gif" alt="Required!" />'; 
    } else {
        name.style.border = '1px Solid Gray';
        namecheck.innerHTML = ''; 
    }
    if (isBlank(pass.value)) {
        pass.style.border = '1px Solid Red';
        passblankAlert = true;
        passcheck.innerHTML = '<img src="required.gif" alt="Required!" />'; 
    } else {
        pass.style.border = '1px Solid Gray';
        passcheck.innerHTML = ''; 
    }
    if (emailAlert) {
        alert("Please enter a valid email.");
        return false;
    } else {
        if (nameblankAlert || emailblankAlert || passblankAlert) {
            return false;
        }
    }
    return true;
}
loginvalidate.js

function isBlank(x) {
    "use strict";
    return (x === null || x === "");
}

function loginRequired(form) {
    "use strict";
    var emailAlert = false, emailblankAlert = false, passblankAlert = false;
    var email = document.forms[form][0], pass = document.forms[form][1];
    var atpos = email.value.indexOf("@"), dotpos = email.value.lastIndexOf(".");
    var emailcheck = document.getElementById("emailcheck");
    passcheck = document.getElementById("passcheck");
    if (isBlank(email.value)) {
        email.style.border = '1px Solid Red';
        emailblankAlert = true;
        emailcheck.innerHTML = '<img src="required.gif" alt="Required!" />';
    } else {
        if (atpos < 1 || dotpos < atpos + 2 || dotpos + 2 >= email.value.length) {
            email.style.border = '1px Solid Red';
            emailAlert = true;
            emailcheck.innerHTML = '';
        } else {
            email.style.border = '1px Solid Gray';
            emailcheck.innerHTML = '';
        }
    }
    passcheck = document.getElementById("passcheck");
    if (isBlank(pass.value)) {
        pass.style.border = '1px Solid Red';
        passblankAlert = true;
        passcheck.innerHTML = '<img src="required.gif" alt="Required!" />';
    } else {
        pass.style.border = '1px Solid Gray';
        passcheck.innerHTML = '';
    }
    if (emailAlert) {
        alert("Please enter a valid email.");
        return false;
    } else {
        if (emailblankAlert || passblankAlert) {
            return false;
        }
    }
    return true;