Bioinformatics Integration Support Contract (BISC)

SYSTEM ARCHITECTURE AND SOFTWARE DESIGN SPECIFICATION

Version 5.0
Period Of Performance: September 30, 2012 to September 29, 2017

Project Sponsor:
National Institutes of Health (NIH)
National Institute of Allergy and Infectious Diseases (NIAID)
Division of Allergy, Immunology, and Transplantation (DAIT)

Prepared by:
Northrop Grumman Corporation
Information Systems Sector
Health IT
2101 Gaither Road, Suite 600
Rockville, MD 20850
(301)527-6400 Main
(301)527-6401 Fax
Jeff.Wiser@ngc.com
Contents

1.0 Introduction ................................................................................................. 3
   1.1 Scope ........................................................................................................ 3
   1.2 Purpose ..................................................................................................... 3
   1.3 Background .............................................................................................. 3

2.0 ImmPort Tools on Amazon Web Services .................................................... 5
   2.1 Open ImmPort .......................................................................................... 5
   2.2 Beta and Development Applications ......................................................... 13
   2.3 Hosted Applications: ImmuneSpace ......................................................... 18

3.0 ImmPort Tools at NIAID ........................................................................... 19
   3.1 Features Table .......................................................................................... 19
   3.2 Hardware Infrastructure ............................................................................ 21
   3.3 Database Architecture .............................................................................. 23
   3.4 Aspera Server ........................................................................................... 23
   3.5 ImmPort 3.0 ............................................................................................. 24
   3.6 ImmPort 2.0 ............................................................................................. 37
### BISC System Architecture and Software Design Specification Version History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>4/29/2005</td>
<td>Initial release of the BISC Phase II System Architecture and Software Design Specification</td>
</tr>
<tr>
<td>1.1</td>
<td>6/1/2005</td>
<td>Completed Sections 3, 11, 14, 15, and 16, including addition of data models to Section 3. Updated sections 12 and 13. Added Section 17. Added acronym list to Appendix A. Added preliminary design and storyboards for future ImmPort versions to Appendix B.</td>
</tr>
<tr>
<td>1.2</td>
<td>12/28/2005</td>
<td>Revised the Purpose section to note that most requirements for the ImmPort system version 1.0 were derived primarily from working with the Population Genetics Program (Section 1.2). Updated assumptions (Section 1.3). Updated references (Section 1.6). Updated descriptions and diagrams of hardware and software architecture (Section 2). Updated the database design to reflect the current database instances and servers across the development environment, test environment, and production environment (Section 3). Updated the previous Experimental and Reference Data Models Description: Supplement to System Architecture and Software Design Specification Version 1.1 and incorporated it into the sections on experimental data and reference data models (Sections 4 and 5, respectively). Added an ImmPort Ontology Browser schema (Section 5). Made minor changes to correct design diagrams in the following sections: 7, 8, 9, 10, 14, 16, and 17. Updated prototypical mock-ups to ImmPort system screenshots in the following sections: 7, 8, 9, 10, 11, 13, 14, 16, and 19. Expanded reference data query use cases and corresponding screenshots (Section 16). Removed preliminary design artifacts related to an upload wizard approach to experimental data (SASDS v.1.1, Appendix B).</td>
</tr>
<tr>
<td>2.0</td>
<td>6/1/2007</td>
<td>Updated section 1.2 Purpose - includes updated information regarding new sections and update details. Updated section 1.4 Key Objectives - updated relevant version information and clarification of objectives. Updated section 2.1 System Architecture Overview.</td>
</tr>
<tr>
<td>2.2</td>
<td>2/6/2008</td>
<td>Updated section 5.0 Reference Data Models. Updated section 3.2.1 Overall Database Design - provided more detail. Updated Table 3-1 ImmPort Database Schemas.</td>
</tr>
<tr>
<td>3.0</td>
<td>10/29/2010</td>
<td>Re-factored the majority of the document to reflect updated data model and definitions, as well as updating the software design section to reduce redundancy and add new functionality.</td>
</tr>
<tr>
<td>4.0</td>
<td>11/01/2011</td>
<td>Updates to reflect changes in admin, flow cytometry, SFVT (Section 19) areas</td>
</tr>
<tr>
<td>5.0</td>
<td>3/24/2016</td>
<td>Rewritten for AWS instance, removal of old code, etc.</td>
</tr>
</tbody>
</table>
1.0 INTRODUCTION

1.1 SCOPE
The scope of the Bioinformatics Integration Support Contract (BISC) is to provide advanced information technology support in the production, analysis, archiving, and exchange of scientific data for a diverse community of life science researchers.

1.2 PURPOSE
The Immunology Database and Analysis Portal (ImmPort) applications have been developed by a team led by Northrop Grumman Information Systems (NGIS) with academic partners from the University of Texas-Southwestern in Phase I, and the University of California San Francisco in the current contract. The ImmPort system is intended to serve as a long-term, sustainable archive of data generated by investigators funded through the Division of Allergy, Immunology and Transplantation (DAIT) of the National Institute of Allergy and Infectious Disease (NIAID), National Institutes of Health (NIH). The ImmPort system consists of an extensive data warehouse containing an integration of experimental and clinical data supplied by NIAID/DAIT-funded investigators. The ImmPort system is freely accessible as a resource to all scientists in the research community.

This System Architecture and Software Design Specification (SASDS) defines the overall ImmPort architecture and software design specification identified by the NGIS BISC Team (hereinafter referred to as the BISC Team) for the ImmPort system that has been developed for NIAID/DAIT. The architecture and design described in this document focuses on the capabilities that are implemented in the ImmPort family of applications as of March 31, 2016.

1.3 BACKGROUND
The key objective of the SASDS version 5.0 is to provide an update to the hardware and software specifications of the system. The ImmPort project has evolved, such that systems are now hosted in a production or near production mode at both the NIAID hosting facility as well as Amazon Web Services (AWS) cloud environment. In general, the long term goal, which is in progress, is to provide hosting of data while it is private and being QC’ed and curated in ImmPort related applications housed at the NIAID hosting facility. When data is shared to the general scientific community, data would be transferred to AWS for easier re-use of these data for analysis, or in short, to bring the data to the analysis tool. This bifurcation of systems allows BISC and NIAID staff to maintain maximum control over data while it is sensitive and private, and more flexibility for re-use and distribution when the data is shared in AWS.

(Note: Some exceptions may occur to this policy, such as the current Data Browser that is hosted at NIAID, which allows downloading of shared research and clinical data. This decision to also host shared data for download at NIAID is driven by the desire to allow an alternative to accruing data transfer costs from AWS.)

Since the last version of the SASDS document, there has been a change in focus in the ImmPort project to lessen the level of effort spent on the development, maintenance and sales/outreach for analysis tools and reference data capabilities. In 2014-2015, based on usage statistics a decision was made in tandem with NIAID program staff to gradually remove analysis tools such as the HLA analysis pipeline, TagSNP, GenePattern and PED data generator as well as the reference data searches (Gene, Protein, HLA, SNP, etc) due to a lack of usage. The FLOCK flow analysis tool suite continued to have constant usage and increased interest and publications related to tool usage and results, so the decision was made to continue support of that application. As a result of these decisions, the retired tools and queries will not appear in this design document.
For the remaining features of ImmPort, it was recognized that a general code refresh was necessary given the overall age of the software and supporting stack of frameworks. As a result, the BISC team has progressed incrementally through the upgrade of the features into a new software architecture that is detailed in this document. For the purposes of this document, the newer code architecture is referred to as "ImmPort 3.0", while the prior architecture being gradually replaced is referred to as "ImmPort 2.0". This document will detail the ImmPort 3.0 architecture for features that have been upgraded or will soon be upgraded, and will keep the existing documentation in place for ImmPort 2.0 features not yet upgraded in the production environment. As features are completed, this document will be accordingly updated during 2016 and 2017. The functional requirements documents for features that are in progress for upgrade will also be available and referenced in this document.

Compared to prior versions of this document that detailed the ImmPort 2.0 architecture, the overall system architecture for ImmPort 3.0 is being simplified to have a less dense middle-tier. As a result, the need to document detailed design packages in this SASDS is reduced, since the same middle-tier approach is utilized across the features in a given application and the EJB tier has been removed.

The database documentation has been moved online, so is no longer described in detail in this document. References to the freely available online materials will be provided in this document. Far more detail about the database fields, tables and ERD diagrams are available online than in prior versions of the SASDS, so the overall amount of information has increased markedly. Additionally, since the SASDS document was released the entire database in MySQL is available for anyone to download and re-use, making comprehension of the database architecture much simpler.

This document is to be considered a “work in progress” and will evolve during the life of the BISC effort as additional requirements are implemented, new requirements are identified, and others are modified or deleted.
2.0 IMMPORT TOOLS ON AMAZON WEB SERVICES

The ImmPort tools deployed on Amazon Web Services (AWS) are designed primarily to identify studies of interest for users to evaluate for future analysis. Currently, the application performing this feature is Open ImmPort. In addition to Open ImmPort, the AWS infrastructure is utilized to develop Alpha and Beta tools to obtain feedback on utility from the user community. Tools such as ImmuneXpresso and the Cell Ontology browser fit into this category. Finally, the AWS infrastructure is being used to host production applications developed by other research teams funded by DAIT without the funding to support a federal system. ImmuneSpace is the first example of this usage of BISC resources.

2.1 OPEN IMMPORT

Open ImmPort is the application utilized by researchers to identify studies of interest for further analysis. As of March 31, 2016, 211 studies have been shared and are cataloged in Open ImmPort. Currently, no row level results are viewable from within Open ImmPort, with the advantage that no user authorization is then required. Open ImmPort utilizes a standard suite of Linux-based AWS servers and components described further below and a Spring/Java and JavaScript web application architecture.

2.1.1 Feature Summary

Table 2.1-1 below summarizes the major functionality of Open ImmPort.

<table>
<thead>
<tr>
<th>#</th>
<th>Capabilities/Features</th>
<th>Capability/Feature Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Search for Studies</td>
<td>Allows users to perform a “Google like” search to identify shared studies of interest. The text entered by the user is searched against an index of the entire set of metadata stored in the ImmPort database.</td>
</tr>
<tr>
<td>2</td>
<td>View/Filter Query Results</td>
<td>After initial search, users are presented with a list of studies, summary information about each study, and the search hits. From here, users may utilize facets on the left hand panel to further filter the studies returned by categories of data such as assay method, species, sample type, etc.</td>
</tr>
<tr>
<td>3</td>
<td>View Study Details</td>
<td>Once a study of interest is identified, users are able to view extensive metadata and summary data about the study.</td>
</tr>
<tr>
<td>4</td>
<td>Visualize Summary Data</td>
<td>For selected aspects of the study data such as demographics, users are able to filter and visualize data in standard plots such as bar charts, line charts by factors such as gender, ethnicity, and arm</td>
</tr>
<tr>
<td>5</td>
<td>Link to download data</td>
<td>For a given study, users can click to download the raw study data, and are redirected to login to the Data Browser application at NIAID.</td>
</tr>
<tr>
<td>6</td>
<td>View Reference and Static content</td>
<td>Static content such as Tutorials, curated cytokine lists, system documentation, user documentation and ImmPort and BISC project information is hosted on the Open ImmPort site.</td>
</tr>
</tbody>
</table>
2.1.1.1 **Home Page**

The Open ImmPort home page provides the entry-point into Open ImmPort. Beyond being an information page containing announcements and ImmPort background information, users can right away begin searching and filtering for studies of interest. In addition, studies that may be of interest to the user community are highlighted in the slider bar based on factors such as recent publications, popularity of the study based on views or downloads, and recent additions to the shared ImmPort content.
2.1.1.2 **Search Bar and Results**

Once the user has selected to view either all or a subset of studies, a list of studies are presented as shown below. The layout of the page is a familiar design with faceted search capabilities illustrated in the left-hand panel, and a “Google-like” simple text search bar at the top. From this page, users may either select a study to view more details, filter the study list further based on facets, click to view a larger version of the study schematic graphic, or click to download the study data. Clicking on the Download button directs a user to the Data Browser application in the directory for that study after authentication.
2.1.1.3 Study Detail Page

When a study is selected, the user is presented with a study detail page. Within this page, there are multiple tabs presented with different aspects of study data based on what has been provided. The full set of tabs that may be displayed include:

- Summary: title, description, PI, type, arms/cohorts, study schematic
- Study Design: study timeline, inclusion and exclusion criteria, schedule of events
- Adverse Event: summary of adverse event data by severity, name, arm
- Assessment: summary of assessments taken
- Interventions: summary of interventions performed
- Medications: concomitant medications taken
- Demographics: summary information about gender, age, ethnicity
- Lab Tests: summary of laboratory panels and tests
- Mechanistic Assays: summary of assays performed, protocols, platforms, reagents, treatments
- Study Files: catalog of study data files provided
2.1.2 Hardware and Software Architecture Components

Table 2.1.2-1 below describes the hardware components and software components that constitute the architecture of Open ImmPort.

<table>
<thead>
<tr>
<th>Component Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model View Controller (MVC)</td>
<td>A software design pattern for implementing user interfaces, often web based applications. The architecture divides the software into three interconnected domains in order to separate data representation from the presentation of the data to the user.</td>
</tr>
<tr>
<td>Business Logic/Services</td>
<td>The controller layer that converts data from the data access tier to the presentation/view tier and executes any necessary business logic.</td>
</tr>
<tr>
<td>Persistence/Data Access</td>
<td>The model tier that manages the connectivity to the data layer.</td>
</tr>
<tr>
<td>Presentation</td>
<td>The view layer that organizes data for presentation to the user.</td>
</tr>
<tr>
<td>RESTful web services</td>
<td>Based on Representational state transfer (REST) architecture and are scalable, light weight and maintainable components often used as web application API's.</td>
</tr>
<tr>
<td>Private subnet</td>
<td>In AWS design, instances in the private subnet cannot receive inbound traffic from the Internet and cannot send outbound traffic to the Internet unless provided through a network address translation (NAT) gateway that is housed in the public subnet. Data servers in the private subnet also access the public subnet through the NAT.</td>
</tr>
<tr>
<td>Public subnet</td>
<td>In the AWS design, instances in the public subnet can receive inbound traffic from the Internet and can send outbound traffic to the Internet. This would be the location of web servers and any NAT servers directing traffic to a private subnet.</td>
</tr>
<tr>
<td>JavaScript Object Notation (JSON)</td>
<td>A lightweight, human-readable data-interchange format widely used in web applications and web-based API's.</td>
</tr>
<tr>
<td>Spring Boot</td>
<td>Provides a ready to start Spring-based application deployment that utilizes best practices for the configuration (<a href="http://projects.spring.io/spring-boot/">http://projects.spring.io/spring-boot/</a>)</td>
</tr>
<tr>
<td>Spring Web MVC</td>
<td>Provides model-view-controller architecture (MVC) and components to develop flexible and loosely coupled web applications in the Spring family of components.</td>
</tr>
<tr>
<td>Aspera Connect Server</td>
<td>Server product developed by Aspera Software that optimizing connectivity speed and reliability over the internet between a client and the server using the proprietary FASP® transport technology. It is widely used for transporting large files, such as sequencing files, over the internet. <a href="http://www.asperasoft.com">www.asperasoft.com</a></td>
</tr>
<tr>
<td>Aspera Connect Client</td>
<td>A free client software package that is a web browser plugin to connect to the Aspera Connect server to optimize connection speed across the internet.</td>
</tr>
<tr>
<td>MySQL</td>
<td>A widely used open source relational database system now owned by Oracle corporation. <a href="http://www.mysql.com">www.mysql.com</a></td>
</tr>
<tr>
<td>Angular JS</td>
<td>A popular environment with interactive components for developing dynamic web-based applications using AJAX/JavaScript developed by Google but open source. <a href="https://angularjs.org">https://angularjs.org</a></td>
</tr>
<tr>
<td>AJAX</td>
<td>Asynchronous JavaScript and XML (AJAX) is the use of XMLHttpRequest object to communicate with server-side scripts, widely used as the primary method for interactive web applications.</td>
</tr>
<tr>
<td>Tiles</td>
<td>Software framework that emanated from Struts allowing developers to create page fragments that are assembled into pages at execution time and also allows for re-use of fragments to reduce redundancies (tiles.apache.org)</td>
</tr>
<tr>
<td>Java Server Pages (JSP)</td>
<td>Framework to assist developers in making dynamically generated web pages based with Java as the backbone language.</td>
</tr>
<tr>
<td>Apache Tomcat</td>
<td>The most widely used open source software implementation of the Java Servlet, JavaServer Pages, Java Expression Language and Java Web Socket technologies. (tomcat.apache.org)</td>
</tr>
</tbody>
</table>
## 2.1.3 Data Architecture

The data for Open ImmPort is stored in two technologies: MySQL relational database and Apache SOLR/Lucene text index. The MySQL relational database maps closely to the Oracle database used to collect the research and clinical data that is housed at the NIAID hosting facility, but de-normalized to optimize query performance. The MySQL database is used when retrieving data for the Study Detail pages shown above. The SOLR server provides the technology for the text search for both the Google bar as well as the faceted search. Indices are generated from the majority of the MySQL database schema plus documents provided by the researchers, such as protocols and Case Report Forms. The SOLR server also provides the capability of “type-ahead” in the search box.

An AWS RDS MySQL server is utilized for Open ImmPort to take advantage of Amazon’s management of the database server to reduce maintenance costs. The SOLR server exists on its own EC2 Ubuntu virtual machine. The MySQL database and the SOLR indices are refreshed completely with each data release, which occurs every 2-3 months.

A full data dictionary and ERD diagrams for the MySQL schema are available online as described below.

The table and column documentation for the Public ImmPort Schema are available [here](#).

Data Model Viewer is available [here](#).

Entity-Relationship diagrams are available [here](#).
2.1.4 Open ImmPort Server Architecture

On AWS, Open ImmPort is separated into its own private network, or VPC. The outward facing Tomcat web server is contained in a public subnet where the site www.immport.org is directed, while the data servers are contained in a private subnet only available through the NAT EC2 server within the public subnet. This separation limits exposure to security threats in the data tier of Open ImmPort. The configuration follows the AWS recommendations described at http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_Scenario2.html
2.1.5 Open ImmPort Software Architecture

Open ImmPort is a standard Java Spring based web application. The client layer utilizes JavaScript AJAX frameworks such as JQuery and Angular to provide interactive graphical user interfaces. The Model, View and Controller uses Spring Web MVC with Tiles/JSP to layout the web pages. Hibernate provides the data and persistence layer to the MySQL relational database via JDBC. Queries against SOLR are run through the web service connecting to the SOLR Tomcat server.
2.2 **Beta and Development Applications**

2.2.1 In addition to the production application environment for Open ImmPort, the BISC project also provides environments for applications that are in “Beta”, defined here as being reviewed by the user community, or in active development to be released in the future for review. These applications in Beta or development have been loosely grouped in what we call “ImmPort Labs”.

2.2.2 **Server Architecture**

The server architecture for applications in ImmPort Labs is described below. The VPC for the Beta applications is described below, and follows the same design pattern as noted for Open ImmPort.

The server architecture for applications that are in development is described below. These applications do not have a data component, and as a result only have a public subnet. Where required, access is limited by IP address filtering.
Public Subnet

- **ImportDevAuth(EC2)**
  - Authorization Server (9443)
    - This is the authentication and authorization server for the ImmPort ecosystem. It will be used by all ImmPort applications which require authentication and will support Single-Sign On.

- **ImportDevGalaxy(EC2)**
  - Galaxy Server
    - This is the Galaxy Server which will support Option 9.

- **ImportDevRack(EC2)**
  - Artifact Repository (0020/9441)

- **ImportDevNat(EC2)**
  - This NAT instance was automatically created when the VPC was created. It is not currently used.

This will be the artifact repository used to build the ImmPort applications.
ImmuneXpresso is a Beta application built under the BISC Contract Option 7 and was built in collaboration with the team at the Shen-Orr lab at Technion (http://shenorrlab.technion.ac.il/). ImmuneXpresso continues the work of the lab in mining PubMed abstracts to determine relationships between cells and cytokines. The index is stored on a standalone Tomcat EC2 and accessible via a RESTFul API. The front-end technology follows the same design pattern as Open ImmPort utilizing SOLR to query for matches to the Cell Ontology and Cytokine Registry available in ImmPort to built query terms to send to the ImmuneXpresso API. This project is in review for utility to the user community.
2.2.4 Ontology Browser

The ontology browser is another Beta application to support the visualization of the Cell Ontology (http://obofoundry.org/ontology/cl.html), which is of great utility in ImmPort data for standardization of cell populations. The browser provides a force directed graph visualization of the ontology, and utilizes the same software stack (JSP, AJAX, SOLR, MySQL) as Open ImmPort to allow for search for Cell Ontology terms with the addition of D3 for the visualization component. We are in the process of integrating the browser into the Open ImmPort application for visualization cell populations identified in ImmPort results as well as putting other ontologies into the browser to evaluate for utility.
2.2.5 Flow Analysis Tools (in development)

For BISC Option 8, our team was funded to perform a refresh on the Flow Cytometry Analysis tools in ImmPort largely based on the FLOCK algorithm. In addition, the science team from UCSF and Stanford proposed an expansion and modularization of the flow analysis tools, as reflected in the diagram below:

![Flow Analysis Diagram]

Further details about the methods being integrated and the steps of the process will be documented as the tool is developed. For the initial development work, we have chosen to utilize the popular analysis workflow engine Galaxy (https://galaxyproject.org/) to optimize modular method and component development and eventual sharing. The use of a publicly available web analysis framework was chosen over direct replacement of existing code in ImmPort because of the existence of open source tools that largely perform the workflow capabilities of queuing, bursting, and chaining methods in a generic way and the large cost to the ImmPort development team of development and maintaining code to perform those workflow capabilities.

Galaxy is an open web based platform that allows development of workflows and pipelines and facilitates reproducible research. We are implementing a version currently on AWS to provide existing users of FLOCK and the flow analysis tools the ability to review the new flow analysis tools for ease of use and utility. The individual methods and pipelines will be shared to the larger community through the appropriate venue, be it Bioconductor for R methods, the public Galaxy server, and/or in public servers of workflow engines such as GenePattern.

In terms of methods, the biggest upgrade in the pipeline will be the integration of the Single-Cell Analysis by Fixed Force- and Landmark-Directed (Scaffold) visualization method developed by the Nolan lab (http://web.stanford.edu/group/nolan/) partially under BISC funding and published in Science in 2015 (http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4537647/).
2.3 Hosted Applications: ImmuneSpace

2.3.1

ImmuneSpace, available at [www.immunespace.org](http://www.immunespace.org) was developed by the team at the Gottardo lab ([http://www.rglab.org](http://www.rglab.org)) at Fred Hutchinson Cancer Center with the team at Labkey Software ([www.labkey.com](http://www.labkey.com)) under funding of the Human Immunology Project Consortium ([www.immuneprofiling.org](http://www.immuneprofiling.org)). Details about the project are available at the ImmuneSpace site, and the architecture in the Labkey product pages. The BISC team provides hosting and basic IT services on AWS for ImmuneSpace in the AWS instances funding by NIAID/DAIT.

It is in discussion that other applications funded by NIAID/DAIT may also be hosted in AWS instances managed by the BISC team in order to make use of the existing team’s experience with AWS and managing federal systems in the cloud.
3.0 IMMPort Tools at NIAID

The NIAID hosting facility has been the home of the production instances of all parts of the ImmPort application since 2011 prior to the AWS applications development. Currently we are in a transition phase where a fraction of the applications are still based on ImmPort 2.0 technologies while the remainder has been transitioned to ImmPort 3.0. It is anticipated that at completion of the software refresh, the majority of the functionality hosted at the NIAID facility will be centered around private research and clinical data submission, QC, and curation before sharing that content to AWS for search and distribution on the cloud. It is also anticipated that the flow cytometry analysis tools will be deployed to AWS to ultimately make use of the burstable cloud capabilities for computation.

3.1 Features Table

Extensive functionality has been built into the ImmPort web application during the duration of the contract period. **Table 3-1-1** summarizes the capabilities and features that have been developed to date.

<table>
<thead>
<tr>
<th>Capabilities/Features</th>
<th>Capability/Feature Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manage ImmPort User</td>
<td>System administration capabilities, which include allowing users to request system access (register), approve registration requests, create user accounts, update user information, query user information, and deactivate users.</td>
</tr>
<tr>
<td>Log In/Off</td>
<td>Authentication and authorization capabilities, which include allowing users to login, logoff, and retrieve login information when the account or password is forgotten.</td>
</tr>
<tr>
<td>Manage NIAID/DAIT Programs</td>
<td>Program management capabilities, which include creating, deleting, modifying, searching, and viewing programs and associating NIAID/DAIT-funded contracts and grants with DAIT programs.</td>
</tr>
<tr>
<td>Manage NIAID/DAIT Contracts and Grants</td>
<td>Contract/grant management capabilities, which include creating, searching, deleting, modifying, and viewing contracts and grants and assigning a PI for a contract or grant</td>
</tr>
<tr>
<td>Manage Research Project (RP) /Private Project Workspace (PPW)</td>
<td>Manage Research Projects (RP), which include allowing a user designated as a PI or PM on a contract or grant to create a project and its associated RP, and update the information associated with the project, manage user access to the RP.</td>
</tr>
<tr>
<td>Manage Collaborative Project (CP)</td>
<td>Manage the CP capabilities, which include allowing a user designated as a PI or PM on a contract or grant to create a CP, update the information associated with the CP, control user access to the CP, and share datasets in a CP.</td>
</tr>
<tr>
<td>Load Experimental Data</td>
<td>Batch loading of experimental data into a RP for multiple types of relevant metadata to provide the minimum information for multiple experimental assay types.</td>
</tr>
<tr>
<td>Browse, Simple Search, Advanced Search, Download Experimental and Clinical Research Data</td>
<td>Experimental and Clinical research data querying, includes allowing users to set up highly advanced multiple attribute search criteria and retrieve details of the experimental data from either a RP or from the semi-public part of the ImmPort database. Query results can be saved in the users RP as lists of entities or results downloaded in multiple file formats. Result files themselves can also be downloaded real-time online, through a queued mechanism, and as a request to receive a physical data delivery.</td>
</tr>
<tr>
<td>Browse, Simple Search, Advanced Search, Export Reference Data</td>
<td>Reference data querying, includes allowing users to set up highly advanced multiple attribute search criteria. Query results can be saved in the users RP as lists of entities or results downloaded in multiple file formats.</td>
</tr>
<tr>
<td>Analysis Dataset Generation and Analysis Tools</td>
<td>Analysis or visualization of reference data and experimental research data from the RP, CP and SPW which includes visualization tools such as GBrowse (Genome Browser), JMOL and Analysis tools including the Flow Analysis Tools, GenePattern for gene expression analysis, TagSNP Analysis Tool, HLA Typing Ambiguity Reduction tools and IGAT. Also allows for the creation of analysis data sets by combining disparate research data including interactive interfaces for creating PED files for genotyping analysis and HLA typing results.</td>
</tr>
<tr>
<td>Capabilities/Features</td>
<td>Capability/Feature Description</td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Database Audit History and Archiving</td>
<td>Maintain a complete audit history of research data (including analysis toolset data created) that is both updated and deleted. The audit history is defined as the ability to capture &quot;who&quot;, &quot;what&quot;, and &quot;when&quot; of the data involved in a change or deletion to Research data contained in the ImmPort System. Additionally, audit and collect limited summary information with respect to auditing/tracking of user session activity on a limited number of database areas. The focus is to obtain summary information on system activity such as logins, information created and updated in the several areas of the Administration Module, and usage of baseline Analysis Tools. Additionally, provide the capability to audit and track user session log information.</td>
</tr>
<tr>
<td>Share Research Data</td>
<td>Provide the capability to subset clinical and experimental metadata from RP areas to be shared into CP through an interactive user interface.</td>
</tr>
<tr>
<td>Browse and Download Shared Research Data</td>
<td>Provides the ability to navigate through a familiar file explorer styled interface to identify study results of interest, to select those result files, and then to download the result files using the Aspera transfer technology.</td>
</tr>
</tbody>
</table>

Development of the administrative functions of ImmPort, including User Management and Authentication, Program administration, Contract administration, and Project administration, are in the process of being refreshed into the ImmPort 3.0 codebase. Requirements for the features are captured in the ImmPort Administration Functional Requirements document that will be available on Open ImmPort by March 31, 2016.

Development of the Research Data Management capabilities in ImmPort for private data is ongoing. These requirements will be available on the ImmPort website within the coming two months as well.

The Flow Cytometry analysis tools are also in the process of being developed, and will be deployed for evaluation to the user community on AWS using the Galaxy framework to manage pipelines. Next steps for the flow cytometry tools will be based on this user feedback.

These functional requirements documents will provide an outline of the capabilities and general user interface descriptions for the given area.
3.2 HARDWARE INFRASTRUCTURE

The NIAID hosting facility is operated and, the encompassing hardware and networking systems and applications, administered by staff from the Office of Computational Infrastructure and Computational Biology (OCICB). The NIH network is operated by the CIT organization at the NIH level. Staff from NIAID or larger NIH operates all standard hosting activities, including OS administration of servers and network management. Our team coordinates with NIAID staff in performing deployments, upgrading the OS on systems, upgrading database software, responding to security issues and alerts, scanning the ImmPort applications for security issues, storage planning, and overall issue resolution.

The overarching server architecture is illustrated in Figure 3.2-1 below.

The pre-existing ImmPort 2.0 servers are identified as the ImmPort application servers (of which there are two) and the ImmPort Database servers (of which there are two). The Aspera Connect application is hosted on the Aspera physical server. The remainder of the ImmPort 3.0 developed code resides on the virtual ImmPort Data Submission Server.

All of the servers with the exception of the database servers reside in a NIAID DMZ to allow external access via defined ports. The database servers are further protected via limited access into a private network.

Specifications for the hardware are not available, but are based on the standard NIAID/OCICB family of servers. Firewall information is not available due to the proprietary nature of the NIH networks.
Figure 3.2-1 Server Architecture at NIAID
3.3 DATABASE ARCHITECTURE

The ImmPort system database architecture is stored and maintained in an Oracle 11g Enterprise Edition database utilizing Real Application Clusters (RAC), installed on a Linux EL5 operating system. Installed database options include Oracle Partitioning, On-Line Analytical Processing (OLAP), and Oracle Data Mining (ODM) options. The RAC environment provides necessary system load distribution and load balancing, while at the same time providing system redundancy and failover capabilities.

In addition to the BCPUB database instances deployed at the NIAID facility, an additional QA database instance is housed at that facility to perform installation testing. Specifications are identical to the production instance.

The data capture schema at NIAID for housing private data is being brought in line with naming schemes and relationships in the "shared" MySQL database available on Open ImmPort. A full data dictionary and ERD diagrams are available online as described below.

Schema Documentation
The table and column documentation for the Public ImmPort Schema are available here.

Data Model Viewer is available here.

Entity-Relationship diagrams are available here.

3.4 ASPERA SERVER

We have integrated technology from Aspera, an IBM company, into the ImmPort system architecture. The Aspera Connect Server using the patented FASP® technology allows for optimized data transfer speeds across the Internet; we have utilized this technology for both data submission and data downloads for large files.

The Aspera security infrastructure provides user authentication and permissions on file systems. A free Aspera Connect Client provides a plug-in for users to install locally to take advantage of the FASP® UDP based transfer optimization. The Aspera SDK is utilized currently for accessing the Aspera Connect Server from the ImmPort application and data submission servers to queue up transfer tasks and return results to users.

More detailed documentation regarding the Aspera Connect Server, the Aspera SDK, FASP® technology, and the Aspera Connect Client can be found online at http://asperasoft.com/software/transfer-servers/
3.5 IMMPORT 3.0

The software architecture for ImmPort 3.0 applications largely mirrors that of Open ImmPort. Some aspects differ based on system requirements, but generally utilize Java, Spring, and AJAX components while making use of the Oracle 11 databases at NIAID.

3.5.1 Hardware and Software Architecture components

The hardware and software components that differ or extend the Open ImmPort components previously described are listed below in Table 3.5.1-1.

Table 3.5.1-1 Software and Hardware Architecture Components

<table>
<thead>
<tr>
<th>Component Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Attached Storage (NAS)</td>
<td>General networked storage providing access to diverse set of clients. Storage specifications are not available but are managed by NIAID OCICB.</td>
</tr>
<tr>
<td>Spring OXM</td>
<td>Spring's Object/XML Mapping support provides marshaling of objects to XML and un-marshalling in the reverse direction for object relational data mapping</td>
</tr>
<tr>
<td>Spring JDBC</td>
<td>Provided by the Spring framework as a value-added JDBC database connection that handles transactions, looping, connection opening and closing, and exception handling</td>
</tr>
<tr>
<td>Oracle RAC</td>
<td>Oracle Real Application Clusters (Oracle RAC) is a clustered version of Oracle Database based on a comprehensive high-availability stack that can be used as the foundation of a database cloud system as well as a shared infrastructure, ensuring high availability, scalability, and agility for any application. (<a href="http://www.oracle.com">www.oracle.com</a>)</td>
</tr>
<tr>
<td>HyperSQL Database (HSQLDB)</td>
<td>Java based database that performs both in-memory and disk-based tables, and is widely used in embedded applications without a large server footprint. (hsqldb.org)</td>
</tr>
</tbody>
</table>
3.5.2 Data Browser

The Data Browser application provides a file explorer interface to users who have identified a study of interest either through Open ImmPort or through knowledge of the study accession. The user can navigate between studies or through directories of protocols, result files of various types, and study data files and select from those directories studies or directories to download. After selection and the click of the Download button, a request is sent to the Aspera server for download of specific files, and the Aspera Connect client coordinates download of the selected items to the user computer. The application requires installation of the Aspera Connect Client into the user’s web browser to download content, but not to view files and directories.
3.5.2.1 Data Browser Software Architecture

The Data Browser is a web-based application with use of similar web frameworks as Open ImmPort described previously. The interactive UI is handled by Angular JS components using AJAX and JavaScript, while the MVC architecture is implemented in Spring. Aspera Security is utilized when calls are made to download content on the data files and directories identified in the Aspera Node Server. The Aspera Connect Server governs the interactions with the data files and directories via the Aspera Connect Client.
3.5.3 Data Submission

Please check that you are using the latest version of the ImmPort data transfer templates.

One of the primary goals of the BISC project is the sharing of research data at a level of detail that is beyond the current published literature dissemination mechanism. The ImmPort data upload module is designed to capture information regarding experimental results and the accompanying metadata, including subjects, biological samples, experiments details, etc. Uploaded information is access controlled by linking to a private research project and the user must be authorized to submit information for this project.

Data Submission Workflow Overview

The ImmPort system has developed a series of templates using Excel to define how information is to be submitted for each entity (https://immport.org/immportWeb/display.do?content=DataTemplates).
Data Submission Templates

Data Submission / Data Submission Home

Welcome to the ImmPort Experiment Data Submission Main Page!

The Experiment Data Submission module allows users to upload experiment data into the ImmPort DATABASE. Information about the experiment and its results are organized into several components including Subjects, Biological Samples, Experiment Samples, Protocols, Reagents, and Experiment Sample Results.

If this is your first time entering data, you will need the following:
- A list of protocol documents delineating your SDM
- Reagent list
- A list of types of samples (blood vs. lymphocytes vs. serum, etc.)
- A system for uniquely defining each sample such that sample type, date of sample, type of analysis can be discerned.

Please check that you are using the latest version of the ImmPort data transfer templates.

- Data Submission Template Change History
- User Guide
- ImmPort Upload Templates Description
- Example Data Packages
- Data Package Validator
  - This tool checks the format and content of the files in the data package and reports issues.
  - The tool is downloadable client side and executed from a graphical user interface.

YouTube Video Tutorials:
- Introduction to ImmPort data loading, part 1 confirming access to your ImmPort project
- Introduction to ImmPort data loading, part 2 basic study design and protocols templates
- Introduction to ImmPort data loading, part 3 creating your zip package
- Introduction to ImmPort data loading, part 4 installing the ImmPort validator on windows
- Introduction to ImmPort data loading, part 5 validating your data submission package

- Minimum Clinical Information For Data Archives
- Subject De-Identification Process

Extensive documentation, including a user guide, example data packages, YouTube video links and full template specifications are available at the ImmPort site.

Template Documentation
The data submission process is described in the User Guide (https://immport-submission.niaid.nih.gov/downloads/documentation/ImmPort_Data_Submission_Guide.pdf)

The ImmPort templates descriptions are available in interactive form here.

The ImmPort templates are available to download and fill here.

The ImmPort template's history is available here.

The ImmPort templates descriptions in PDF form are available here.

Data Model Viewer is available here.

Excel templates are used to annotate columns and provide features such as lists of controlled vocabulary terms. The research data content is organized into generally understandable biological domains including
study, subjects, samples, protocols, reagents, and experiments. There is a template for each domain. The records in one domain often reference user defined records in other domains (e.g. subject records reference protocols for subject assessment and treatment). The data upload process supports a modular or incremental approach so that subsets of a study’s content may be uploaded to ImmPort at the discretion of the data provider. Data uploaded in an earlier upload session may be referenced in a subsequent session without having to redefine the records. The data uploaded by users may be referenced by their own identifiers or by ImmPort accessions. User defined identifiers must be unique within a research project. ImmPort accessions are unique across the data repository. Each upload template has required fields that represent the minimum information that must be submitted (as defined by the NIAID DAIT minimum information guidelines - https://immport.org/tutorials/MinimumInformationGuidelines.doc), plus additional fields that are optional. In some cases, references from a record in one domain to a record in another domain are required and in other cases they are optional. This establishes links between uploaded data and ensures that the data model is effectively populated.

A commonly adopted practice is for the data provider to upload their data when their study is completed. The user fills out one or more templates. If the templates are in the Excel format, they saved from Excel into tab-delimited text files. This eliminates the need to support multiple versions of the Excel spreadsheets. These templates are bundled together into one ZIP archive with additional documents such as protocols and assay results. The ZIP archive is submitted to ImmPort via web based file upload technology. This generates an upload ticket that is used to track status and audit uploaded data. After the package is submitted, the system displays a confirmation screen and the package is queued for later processing.

Aspera transfer technology is again utilized as an option in data submission to optimize the speed of transfer. The user is required to have installed the Aspera Connect plug-in to perform a submission in this manner.

There are alternate procedures to deal with large data upload requirements that do not scale well with web-based technology. An upload ticket may be generated without uploading a ZIP archive. This ticket is used to link data transferred to the data upload staging area via non-web based means. For large data sets, an external drive may be sent to a data provider who uploads data to the drive and then sends it back to ImmPort. The data is transferred to the ImmPort repository and the standard data processing that parses and links the data to the content in the repository is completed. In some cases, a hybrid approach is taken whereby descriptive data is uploaded via the web interface and additional data is sent to ImmPort by offline methods.

**Data Submission Processing Workflow**

When a data package is uploaded, the system assigns a ticket number and records an entry in the upload registration table for this package. The batch upload daemon runs every 5 minutes and checks the database registration table looking for packages to process. When the upload daemon finds a package to process, it does the following:

1. Marks the package as processing
2. Unpacks the package in a working directory
3. Parses each template in predefined order, because there can be dependencies between the templates. For example, every biological sample record must link to a subject record, so the subject template, must be loaded before processing the biological sample template
4. As each template is parsed, the records are inserted into the ImmPort database.
5. Any file in the package that do not match the template names, is treated as a file to be archived and linked to the project.
6. If the package is successfully loaded, the user is notified that the submission was completed successfully. If any error occurs during the load, the entire package is rolled back and the user is notified of the errors.

**Data Submission History**

The user may review the data submission history to their project(s) to get a status update and a log of their data submission sessions.
This page is a summary overview of the data submissions to a project. If you have access to more than one project, use the project filter to view the submission queue from other projects.

The data submitted to iMnPort can be queried and reviewed in the Research Data Search.

**Filter By Project:** Public Genome Expression Data

Click on a Ticket Number below to view a submission's details.

<table>
<thead>
<tr>
<th>Ticket Number</th>
<th>.ZIP File Name</th>
<th>Format</th>
<th>Status</th>
<th>Submitter</th>
<th>Submit Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bea_20090904_11166</td>
<td>HLA_Tregertperlocc_Esperiment_With_delimited.zip</td>
<td>HLA_typing_summary</td>
<td>Completed</td>
<td>Bea</td>
<td>05/04/2009 10:51:23</td>
</tr>
</tbody>
</table>
3.5.3.1 **Data Uploader Software Architecture**

The data submission “Uploader” does not consist of an interactive user interface since it relies on files loaded into a specified directory and a queue established in the database. The software architecture has been upgraded to the ImmPort 3.0 stack, and utilizes the Spring MVC, object relational mapping, and database connection frameworks.
3.5.4 Data Submission Validator

To assist the user with checking the validity of the templates they have filled out, ImmPort has developed a stand-alone Java application (DataValidator) that can be downloaded and run on the user’s desktop before they upload the ZIP archive for processing. The DataValidator checks content for referential integrity between the templates, controlled vocabulary fields, previously assigned Project based identifiers, etc. Using the DataValidator is optional, but if used helps find most of the common data submission errors, before the package is submitted for final processing.

The Validator uses a HSQLDB database that is populated from content downloaded by the user via a RESTful web service connected to the ImmPort database. The HSQLDB database stores a subset of the content in the project necessary to validate referential integrity. The Validator synchronizes with the ImmPort repository automatically via the web service. After synchronization, the user selects the data upload archive to evaluate and the Validator provides a report on the success of the validation or issues it encountered.
3.5.4.1 **Data Validator Software Architecture**

The Data Validator software has been upgraded to the ImmPort 3.0 stack. One overriding difference between the Data Validator and the rest of the ImmPort 3.0 applications is that the Data Validator is installed on a user’s computer directly. This allows the user to be disconnected from the Internet and ImmPort to perform basic validation, and additionally, saves the effort of submitting large packages to ImmPort before knowing that there is an issue. The entire embedded Tomcat Server is installed in a package on the user machine via a multi-step installation wizard, which works for Windows, Mac, and Linux based computers. Having the entire web server installed on a remote site opens the possibility that users could install the Validator on a central location and having multiple users access the application.

Again, AJAX components are utilized for the interactive UI, and the Spring Web MVC handles the web tier model, view and controller. The Web Service is utilized for user authentication and retrieval of projects to which the user has access. Following authentication and project selection, the Web Service retrieves a subset of the submitted data in the ImmPort database (namely identifiers for each submitted object) that is used within a session of the Data Validator to validate local data packages. A copy of the data content retrieved from the web service is also stored locally, so a user may operate in off-line mode. The identifiers are stored within the session into the HSQLDB, which offers a high performance in memory database that opens at run-time and closes completely at session close.

Having the installed Data Validator in a similar web application framework as the rest of ImmPort 3.0 provides substantial benefits in code re-use and maintenance versus the prior iteration of the Data Validator that was implemented in Java Swing. Usage of HSQLDB for the validation steps, allows consistent use of SQL for running validation queries between the code executed in the Data Validator and the Data Submission codebase.
3.5.5 Curation Application
The Curation Application is a relatively simple web-based application utilized as an alpha tool by the internal scientists and data curators on the BISC team in Northrop Grumman. The MVC architecture is the same as the previously highlighted applications. The primary function of the Curation application as currently developed is to assist curators in executing and saving database SQSL queries that help with daily operational tasks in getting data collected, QC’ed, shared and curated. Based on usefulness to the internal curation team, it can be evaluated over time if the application would also be useful for external users.

3.5.6 Downloads Application
The Downloads application provides a means to host static content with no other purpose. No authentication is required. The goal is to move all static documentation and files used by the web applications in the NIAID facility into this application.

3.5.7 Support Server
The support server listed in the hardware diagram is a development server used for trying out computational and data generation tasks. There is no web application installed on this hardware and it is not available to the outside world other than through VPN access to the NIH network by badged employees.
3.6 IMMPort 2.0

The IMMPort 2.0 system is implemented as an n-tier distributed architecture, providing a scalable, pluggable, and reusable componentized solution where additional layers or components can be added to accommodate current and future needs. The architecture is comprised of BigIP F5 hardware load balancers, a pair of application server nodes running the Redhat Enterprise Linux 5 OS, a 2-node Real Application Cluster (RAC) Oracle database, and a Network Attached Storage (NAS).

The software organization is implemented using a conventional 3-tier approach, which decouples software components into three major areas: the presentation, the business, and the persistence layers. Each layer specializes in specific orchestrated functions. The presentation layer separates the Graphical User Interface (GUI) from the complexity of the business rules. The persistence layer abstracts data retrieval and storage from yet another complex layer such as that of a Relational Database Management Systems (DBMS).

The presentation layer utilizes key web technologies, included but not limited to, the Hypertext Markup Language (HTML), Extensible Markup Language (XML), Cascading Style Sheets (CSS), Java Server Pages (JSP), Asynchronous JavaScript and XML (Ajax), Extjs, and the Struts Framework. The business tier leverages the Stateless Enterprise Java Beans (EJBs) and the Message Driven Beans (MDBs) stack of the Java2 Enterprise Edition (J2EE) framework. Using the Facade Object Oriented Design pattern, the EJBs abstracts the complexity of the underling complex business rules. Lastly, the persistence layer manages all database access and storage and it relies on the successful object-relational mapping (ORM) Hibernate library for the Java platform.

3.6.1.1 Software Architecture

The IMMPort system software architecture diagram shown below depicts all the components comprising the BISC software architecture that have not yet been upgraded and that are housed on the IMMPort Application Servers and the Database Servers previously described. Each software component is described in Table 3.6.1-1, which follows the figure.
Figure 3.6.1-1: Software Architecture
### Table 3.6.1-1. Software Architecture Components

<table>
<thead>
<tr>
<th>Component Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Server</td>
<td>The Application Server contains JBoss Application Server instance(s) and an Apache Web Server. The JBoss App Server is responsible for hosting the presentation tier, business logic tier, and the data access tier. For purposes of performance and scalability, multiple application server instances may be utilized to support running a cluster of JBoss application servers.</td>
</tr>
<tr>
<td>Business Tier</td>
<td>The business tier abstracts the complexity of the business rules and manages the communication between the presentation and database tiers.</td>
</tr>
<tr>
<td>Database Servers (2 nodes)</td>
<td>This software component contains two major database areas: Public Data Warehouse and Private Project Workspace. There is also a Raw File Repository for staging and backup/archive purposes. To ensure proper performance and scalability, there will be at least two of these servers running Oracle 10g in Real Application Cluster mode.</td>
</tr>
<tr>
<td>EJB Container</td>
<td>The application tier consists of the Enterprise Java Bean (EJB) container and the Java Messaging Service. The EJB container houses the majority of the application logic in terms of data loading/publishing, browsing/querying, journaling, and any administrative/management activities. The JMS component, on the other hand, is responsible for the execution of any application logic that is asynchronous in nature. This helps reduce the load and distribute the work across the network.</td>
</tr>
<tr>
<td>Hibernate</td>
<td>The data access tier utilizes the open source Hibernate library for data access. Hibernate provides fine-grained access/update, caching, and transactional persistence to the Private Project Workspace database where transactional support is required.</td>
</tr>
<tr>
<td>Java Messaging Service</td>
<td>Java Messaging Service (JMS) is the standard API for sending and receiving messages.</td>
</tr>
<tr>
<td>Java Server Pages</td>
<td>Java Server Pages (JSP) is server-side technology for building presentation tier components.</td>
</tr>
<tr>
<td>JBoss Application Server</td>
<td>JBoss Application Server is an open-source J2EE application server.</td>
</tr>
<tr>
<td>Oracle 11g Database</td>
<td>Oracle 11g is a relational database management system (RDBMS). This software component contains all subcomponents that make-up the server-side RDBMS software. This includes, but is not limited to: Cluster Ready Services (CRS) software, Real Application Clusters (RAC) software, and RAC database server software. The server configuration is a 2-node RAC environment with shared Automatic Storage Management (ASM) utilizing ASMLib software.</td>
</tr>
<tr>
<td>Oracle 11g Client</td>
<td>This software component contains all subcomponents and utilities that are installed as part of the Oracle 10g Client software, currently residing on the ImmPort Application Server. This SQL*Plus Client subcomponent is used by the GBrowse application software for establishing a remote client database connection for retrieval of data for the GBrowse Visualization Tool.</td>
</tr>
<tr>
<td>Persistence Tier</td>
<td>The persistence tier manages all database access and storage using the open source object relational mapping Hibernate library.</td>
</tr>
<tr>
<td>Presentation Tier</td>
<td>The presentation tier manages and renders Graphical User Interface (GUI) components using web technologies such as HTML, CSS, JSP, XML, Ajax, and the Struts framework.</td>
</tr>
<tr>
<td>Struts</td>
<td>Struts is an open-source Model View Controller (MVC) framework for developing J2EE web applications.</td>
</tr>
</tbody>
</table>
3.6.2 User Management and Authentication

The Manage User design artifacts model system administration capabilities, which include allowing users to request system access (register), approve registration requests, creating user accounts, updating user information, querying users, and deactivating users from the system.

Create/View/Update/Search User

The Create/View/Update/Search User depicts the events for creating/viewing/updating/searching a user account. The system collects the user information and creates a user account, provided the submitted data are valid. If the system encounters errors, then it will display the appropriate message. Once the user information is collected the system will display the data while giving the user an option to update the information.
Create New User Class Diagram

- ProcessSelectCommunityAction
  - + execute()

- DisplayCreateUserAction
  - + execute() (null)

- ProcessCreateUserAction
  - + createUser()
  - + execute()

- DisplayViewUserAction
  - + execute()

- UserManager
  - + createUser()
  - + getUserData()

- UserManagerDAO
  - + getUserById() (null)
  - + save() (null)

- ResearchProjectManagerDAO
  - + insertResearchProject() (null)

- ContractGrantManagerDAO
  - + insertContractGrantInfo() (null)
Create New User Sequence Diagram
3.6.2.1 **Users Registration**

The User Registration depicts the events for requesting access to ImmPort. First, the individual initiates a registration request.

Depending upon the type of access requested, the system prompts the individual for additional information. The user may select to be a general member of the life sciences community, a member of the DAIT-funded research community, or an extramural NIH employee.
The system validates the user information and saves the registration request, provided the data submitted are valid.

Appropriate error messages are displayed if errors are encountered. At the end of the process, the system notifies those users responsible for reviewing registration requests.
User Registration Class Diagram
User Registration Sequence Diagram

- User accepts agreement
- User selects user community
- User forward to Display.CreateUserAction
- User create user page displayed
- User add user
- User process registration agreement action
- User process select community action
- User process create user action
- User save [Status = Pending]
- User save
- User user added
- User contract grant manager DAO
- User contract grant manager DAO
- User contract grant manager DAO
- User contract grant manager DAO
- User contract grant manager DAO
- User success/failure
- User success/failure
- User success/failure
- User success/failure
3.6.2.2 **Approve/Reject User**

The Approve/Reject User depicts the events for approving/rejecting a user account. Appropriate error messages are displayed if errors are encountered.

---

**Approve/Reject User Class Diagram**

---

**Approve/Reject User Sequence Diagram**
3.6.2.3 **Deactivate User**

The Deactivate depicts the events for deactivating a user account. Appropriate error messages are displayed if errors are encountered.
Deactivate User Class Diagram

Process

+ execute()

UserManager

+ deactivateUser()
+ preDeleteCheck()
...

UserManagerDAO

+ updateUser()
...

User Class Diagram
3.6.2.4 **Activate User**

The activate user depicts the events for activating a user account that has been deactivated in the past. Appropriate error messages are displayed if errors are encountered.
Activate User Class Diagram

Activate User Sequence Diagram
3.6.2.5 **Search Users**

The search user depicts the events for searching a user account. The user is first presented with a search screen.

After the search, results are presented and the user may click to view more details about the selected user.
Search Users Class Diagram

Search Users Sequence Diagram
3.6.2.6  **Forgot password**

The Forgot Password Sequence Diagram depicts the sequence of events for a user that has forgotten their username/password.

A security question is utilized to verify the user’s identity, and the password is emailed upon success.
Forgot Password Class Diagram

Forgot Password Sequence Diagram
3.6.2.7 Log in/Off

In order to log into the ImmPort system, a unique user-id is required with a password. These are created during the registration process or creation of an account. The password is restricted to 8-12 characters with one being a numeric value. Appropriate error messages are displayed, if applicable.
User Administration Class Diagram
3.6.3 Program, Contract, Project Management

ImmPort 2.0 vs. 3.0, etc.

3.6.3.1 Create Program

The create program diagrams depict the events for creating a program. Appropriate error messages are displayed if errors are encountered.

Once the program is created, contracts are associated.
Create Program Class Diagram

Create Program Sequence Diagram
3.6.3.2 **Update Program**

The Update Program diagrams depict the events for updating a program. Appropriate error messages are displayed if errors are encountered.

**Update a Program Class Diagram**
Update a Program Sequence Diagram
Search Program

The search program diagrams depict the events for searching a program. Appropriate error messages are displayed if errors are encountered.

A list of program search results is displayed. This allows the user to select a program to view or modify. In order to modify the search, click the “Modify Search” link.
Search Program Class Diagram

Search Program Sequence Diagram
Program Administration Class Diagram

SessionServiceBase
- ejbCreate()
- setSessionContext()
- ejbRemove()
- ejbActivate()
- ejbPassivate()
- checkProjectMembership()
- checkProjectMembership(
- checkProjectOwnership()
- checkProjectUsers()
- checkCMEMembership()
- checkPM()
- getOwner()
- getLoggedInUser()
- validateUser()
- beginQueryAudit()

HibernateBaseDAO
- save()
- update()
- mergeChanges()
- lock()
- get()
- delete()
- saveOrUpdate()
- batchInsert()
- batchUpdate()
- batchInsertWithSession()
- batchUpdateWithSession()
- batchInsertNoFlush()
- batchInsert()
- batchUpdate()
- setQuerySnapshot()
- setQuerySnapshot()
- getFirst
- getNext
- isProgramTitleUsed()
- isProgramTitleUsed()
- getNextValue()
- distinctValueOf()
- closeConnections()
- setIDDate()
Add/Update Contract

The Add/Update Contract diagrams depict the events for adding and updating a contract. Appropriate error messages are displayed if errors are encountered.

Once the contract is added, a Principal Investigator, Program Manager and Other Staff is associated with the contract.

Once the associated users are added, a default project may be created. The associated users to the contract will obtain access to the default project.
Add/Update Contract Class Diagram
Add/Update Contract Sequence Diagram

1. Request to create new contract
2. Create contract page displayed
3. Add contract
4. Forward to Load/Update Contract Grant Action
5. Update contract page displayed
6. Update contract
7. Contract updated

Actions:
- LoadCreateContractAction
- CreateContractGrantAction
- InsertContractGrantInfo
- GetContractGrantInfoById
- GetContractGrantInfoByCriteria
- UpdateContractGrantInfo
- MergeChanges
- Save
- CriteriaUniqueResult
- ContractInfo

Success/Failure:
- Success
- Failure
**Search Contract**

The Search Contract diagrams depict the events for searching a contract. Appropriate error messages are displayed if errors are encountered.

A user is able to search on the fields Title, Grant – Contract Number, Category, Start Date, End Date, ImmPort Access End Date, Principal Investigator, Associated Program Name or Status.

A list of Grants - Contracts search results is displayed. This allows the user to select a Grants - Contracts to view or modify. In order to modify the search, click the “Modify Search” link.
Search Contract Class Diagram

Search Contract Sequence Diagram
Add/Update Projects

The Add/Update project diagrams depict the events for adding and updating a project. Appropriate error messages are displayed if errors are encountered.

In order to create a project, a user must read the terms and conditions for creating a Private Project Workplace.
The user enters the required fields, Title, Description, Keywords, Category, and Contract-Grant. The Project Owner is auto generated. An appropriate error message will display if an error has occurred.

Once the project is created, the user is able to update the project and/or associate users to their project.
Adding/Updating A Project Class Diagram
Adding/Updating a Project Sequence Diagram
**Display Projects**

The display projects diagrams depict the events for viewing a project.

![My Research Projects Table]

A user is able to select the project while viewing the project details or manage project data.

**Display Projects Class Diagram**

![Display Projects Class Diagram]
Display Projects Sequence Diagram

USER
request to see projects
getAllUserProjects()
getMyProject()
query list()

List of projects are put into 2 lists: RP and CP. Inside a Hash

hash map
list

results displayed

:DisplayMyProjectsAction :ResearchProjectManager :ResearchProjectManagerDAO :Persistence
Project Administration Class Diagram
3.6.4 Flow Cytometry Analysis

Flow cytometry analysis defines and quantifies cellular features making it an essential tool employed in immunology research. Challenges inherent in the management and interpretation of flow cytometry data led to the development of the ImmPort Flow Cytometry Analysis tool and its four supporting modules:

1. Data Management—single/multiple file upload, data set creation/editing
2. FCS file analysis using the FLOCK algorithm
3. Cross Sample analysis
4. Viewing/editing analysis results

The Data Management module allows the user to upload individual or multiple flow cytometry files, edit the descriptive content of the files and create/edit sets of uploaded files. To further facilitate data management and data set creation ImmPort employs components of the search module to find files of interest.

The FLOCK (FLOw Clustering without K) algorithm was developed as a novel method to identify unique populations within a single flow cytometry file. Analysis output includes two dimensional images of FLOCK-defined populations enhanced by the inclusion of the following tables for viewing or download: results summary, centroid, mean fluorescence intensity. (FLOCK publication: http://www.ncbi.nlm.nih.gov/pubmed/20839340).

Cross Sample analysis automatically maps populations of cells across multiple flow cytometry files utilizing centroid values saved from previous FLOCK results. Analysis results are displayed in two-dimensional format for each file in the comparison allowing the user to select which population/fluorescent marker to view across all files. Population proportions are available for all files in the comparison analysis as are results for individual files including centroid and mean fluorescence intensities.

Viewing and editing analysis results applies to FLOCK and individual results in Cross Sample Comparison. The Result Adjustment System (RAS) allows the user to edit two-dimensional images via moving, splitting or deleting a centroid for improved representation of the analyzed results. Supporting statistical output is recalculated to support the analysis edits.

The ImmPort Flow Cytometry analysis workflow is depicted below:
Data Management: File Upload

The output from a flow cytometer is typically a binary file that conforms to the Flow Cytometry Standard (FCS). Most analysis algorithms including FLOCK require the FCS binary file be converted and possibly transformed into a text file representation of the fluorescence expression values. The ImmPort Data Management File Upload module supports both single and multiple FCS file upload. Uploaded files are automatically converted, transformed and stored in the ImmPort database. The conversion algorithm, FCSTrans, was written using R/BioConductor modules by the BISC team as described in https://www.immport.net/help/ImmPort.FCS.conversion.pdf. The ImmPort system supports linking of FCS files to metadata such as subject, study and experimental information, previously loaded into the ImmPort system.

Upload Class Diagram

![Upload Class Diagram](image-url)
Upload Sequence Diagram
Data Management: Create Analysis Set

The Data Management module facilitates the creation of analysis sets by employing existing search module code and technologies outlined in the search module portion of this document. Created analysis sets may be used as input to either FLOCK or Cross Sample Analysis while existing data sets may be edited to add or remove FCS files. The UI consists of 3 panels:

- Search—used to query for available FCS files
- Results—displays search results
- Create Set—files of interest are moved to this panel to save the set to the database.

The meta information for the analysis sets is stored in the ImmPort database.

Create Analysis Set Query Class Diagram
Create Analysis Set Query Sequence Diagram

Create Analysis Set Save Class Diagram

Create Analysis Set Save Sequence Diagram
**FLOCK Analysis**

FLOCK analysis uses a rapid binning approach to determine the number of unique populations in high dimensional flow cytometry data. The user interface provides two means to initiate a FLOCK analysis:

- Individual file selection—single or multiple files can be selected for FLOCK analysis from the Uploaded Flow Files Search screen accessed via FLOCK/Analyze Individual Files dropdown menu
- Data set selection—previously created data sets can be selected for FLOCK analysis via FLOCK/Analyze Data Sets dropdown menu

From either analysis initiation point the process for running FLOCK requires the same steps: create an analysis task with the data set or selected individual files, select analysis settings, place the task in a queue, the FlockWorker retrieves the task and a Java program is called for the following:

1. Extracting the text version of the FCS file from the database and storing it in a task folder.
2. The FLOCK program written in ‘C’ reads the extracted FCS file and generates detailed and summary results representing the cell populations.
3. The result files are processed by the Java program generating overview images.

The FlockWorker process marks the task as completed making the results available for user review and subsequent editing.

**FLOCK analysis: individual file selection**

![Image of FLOCK analysis interface](image)
**FLOCK analysis: data set selection**

![FLOCK analysis: data set selection](image)

**FLOCK Analysis Setting**

![FLOCK Analysis Setting](image)

**FLOCK: Analysis Status**

![FLOCK: Analysis Status](image)
Cross Sample Analysis

Cross Sample analysis automatically maps populations of cells across multiple flow cytometry samples and computes the summary statistics for downstream analysis. The analysis requires a previously created data set and saved centroid file which are presented for user selection in the Cross Sample Analysis UI. The initiating step for Cross Sample Analysis is the creation of an analysis task comprised of the data set and centroid which is followed by a review of the analysis components via Cross Sample Analysis Settings display. The task is placed in a queue where the CrossSampleWorker process picks up the task and calls a Java program for the following:

1. Extracts the text version of the FCS files from the database and stores it in a task folder.
2. The centroid adjustment program written in ‘C’ reads the extracted FCS file and generates detailed and summary results representing cell populations.
3. The result files are processed by the Java program generating overview images.

At the end of the CrossSampleWorker process the task is marked as completed providing the user with access to results which include summary statistics and marker by population two-dimensional displays.
Cross Sample Analysis: Data set and centroid selection

Cross Sample Analysis: Analysis Setting
Cross Sample: Analysis Status

Flow Cytometry Analysis (Beta) / Cross Sample / Analysis Status

Your Cross Sample Analysis request was submitted successfully. Your task ID is 2618. You will be sent an email when your task has been completed.

Click the "Show Analysis Results" button to proceed to the Cross Sample Analysis Detail page.

Show Analysis Results

Submit Cross Sample Analysis task Class Diagram

Submit Cross Sample Analysis task Sequence Diagram
**FLOCK Analysis Results**

Results of FLOCK analysis tasks are displayed in a tabular grid. Each row in the grid represents one task and contains summary information such as name, description, date completed, etc. Selecting Detail provides greater task detail displayed in two panels

- FLOCK Analysis Task Detail—task ID, name, description, etc.
- FLOCK Analysis Result Files—individual FCS file results with a link to individual results detail

The individual file results contain several panels enabling the user to review both statistics and two dimensional images generated by the analysis.

**FLOCK Analysis Results: tabular view**

![Flow Cytometry Analysis (Beta) / FLOCK / Analysis History](image)

<table>
<thead>
<tr>
<th>Details</th>
<th>Name</th>
<th>Description</th>
<th>Task Type</th>
<th>Version</th>
<th>Status</th>
<th>Run By</th>
<th>Start Date</th>
<th>End Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detail</td>
<td>20100931 test 2</td>
<td>flock</td>
<td>2</td>
<td>completed</td>
<td>Yuen</td>
<td>2010-08-31</td>
<td>2010-08-31</td>
<td></td>
</tr>
<tr>
<td>Detail</td>
<td>3 column test</td>
<td>flock</td>
<td>1</td>
<td>completed</td>
<td>Thomson</td>
<td>2010-07-07</td>
<td>2010-07-07</td>
<td></td>
</tr>
<tr>
<td>Detail</td>
<td>test TT0454</td>
<td>flock</td>
<td>1</td>
<td>completed</td>
<td>Yuen</td>
<td>2010-07-07</td>
<td>2010-07-07</td>
<td></td>
</tr>
<tr>
<td>Detail</td>
<td>test edited meri r</td>
<td>flock</td>
<td>1</td>
<td>completed</td>
<td>Thomson</td>
<td>2010-07-07</td>
<td>2010-07-07</td>
<td></td>
</tr>
<tr>
<td>Detail</td>
<td>test overview disp</td>
<td>flock</td>
<td>1</td>
<td>completed</td>
<td>Thomson</td>
<td>2010-07-07</td>
<td>2010-07-07</td>
<td></td>
</tr>
<tr>
<td>Detail</td>
<td>20100727 TT0731 run-run4testapp2</td>
<td>flock</td>
<td>1</td>
<td>completed</td>
<td>Yuen</td>
<td>2010-07-07</td>
<td>2010-07-07</td>
<td></td>
</tr>
</tbody>
</table>
**FLOCK Analysis Results: Detail display**

![Image of FLOCK Analysis Results]

**FLOCK Analysis Results Class Diagram**

![Class Diagram Image]
FLOCK Analysis Results Sequence Diagram
**Cross Sample Analysis Task Results**

The results of Cross Sample Analysis tasks are displayed in a tabular grid similar to that for FLOCK. Each row in the grid represents one task and contains summary information such as name, description, date completed, etc. Selecting Detail provides greater task detail displayed in three panels:

- Task detail—task ID, name, description, etc.
- Flow Analysis Sets—displays all sets in the analysis
- Cross Sample Analysis Result Files—displays all FCS files in the analysis

Selection of Analysis Results in the task detail panel links to the population proportion table and the marker-by-population two dimensional displays. The Results link within the Cross Sample Analysis Result Files panel links to individual file results.

**Cross Sample Analysis: tabular view**

![Cross Sample Analysis Task Results Table](image-url)
Cross Sample Analysis: Detail display

Cross Sample Analysis: Marker by Marker display
Cross Sample Analysis Task Results Class Diagram

Cross Sample Analysis Task Results Sequence Diagram
**FLOCK Results Visualization**

The FLOCK analysis Detail result screen includes 6 displays:
- Overview
- Summary Tables
- Download Tables
- Download Results
- Centroid
- 3D View

Each of the six sections will be reviewed but given the importance of image generation for the Overview and Result Adjustment sections a brief image generation overview is required.

FLOCK analysis produces a tab-delimited text file where rows are events and columns are markers—an exception is the last column which represents the population assigned to the event. The output file is transformed into binary format to facilitate real time image generation. UI requests for images results in a JAVA module reading the binary file, computing the coordinates and drawing dot-plot images in PNG format. Images are written to the file system under the analysis task folder where they are retrieved by the UI.

FLOCK results Overview displays a table of images where each cell represents the dot-plot of the populations assigned based on the expression values for two markers. On the right hand side of the screen there is a panel for selecting which populations to display, and summary information for each population. The user can select/deselect populations using the check box and then click on the Update button to change the display of the images to only include the selected populations. Each image may be selected—this action takes the user to the Result Adjustment screen.
Result Adjustment

Result Adjustment supports editing results as a means to fine-tune the centroid locations. Images will be regenerated when Save Changes is selected. In addition to moving a centroid the system supports splitting and deleting centroids.
Summary Tables

Three summary tables are available for each FLOCK analysis and include:

- **Results**—populations appear as table rows, channels/markers as columns with the population proportion presented in the last column
- **Centroid**—each cell of this table represents the center for the population/marker combination
- **Mean Fluorescent Intensity (MFI)**—each cell represents the mean fluorescence value.

Download Tables

The content of the download tables is the same content that is displayed in the Summary Tables mentioned above. In this case the content can be downloaded in text file format for further processing.
Result files available for download include:
- Results table including event number and population assignment
- A package of files representing all the result files.

**Centroid**

From this panel the user can select to save the results editing from Result Adjustment or undo any previous adjustments.

**3D View**

In this view the results for 3 of the markers can be displayed in a 3D view using the JMOL applet. On the right side panel, the user can select which markers to display and the user can choose which populations to display.
Flow Analysis Class Diagrams

Actions
DAO
3.6.5 Data Management

The search module was designed to allow the user to query for summary information about objects by entering ad-hoc queries, based on the properties of each class of object, plus associations between objects. The module was used to provide support for both advanced search and basic search UI’s. Examples of the types of objects represented in ImmPort are: subjects, experiments, experimental results, studies, genes, proteins, etc.

During the design process we had several goals and constraints that are outlined below:

- Develop the system to be easily maintained.
- Develop the system to be extensible to other objects within the ImmPort schema.
- Reuse the tabular result page design already developed.
- Reuse the Hibernate object model.
- Make much of the system configurable or meta-data driven.
- Support the use of previously saved query results (lists) in the complex queries.
- Keep the interfaces between the layers clean and simple, allowing reuse in other parts of the application.

Below is a screen shot of the Advance Search module for Research Queries. The screen has 4 major components that will be discussed on more detail later in the document.

- Toolbar.
- Query Tree Panel.
- Criteria Panel.
- Summary Results Panel

This query and UI framework has been implemented for basic and advanced research search interfaces and for advanced reference query in the same manner. It is also used in selected queries across the system when it is useful to return a subset of objects based on a criteria specific to a given page. This framework re-use allows for greater maintainability and expedites the addition of new search interfaces to the system.
3.6.5.1 **General Overview**

Multiple search screens provide the entry point into the search framework described, but all utilize the same presentation layer technology. The Advanced Search research page appears with a tree view with available search attributes on the left side, the selected search attributes on the right. The user is prompted to return the type of object that they wish to return at the top.

Basic Search screens have a fixed and minimal set of attributes with which to query for a fixed object to return, as show for the Subject Search page.
After a search is submitted, the attribute panels disappear and the query results summary grid appears. This result summary page provides extensive functionality, including:

1. Sorting by each column
2. Re-ordering of columns by drag and drop
3. Hiding and Un-hiding columns
4. Page through result records
5. Selection of individual object records to view further details
6. Save Items or Save All Item rows to a list for later re-use
7. Export of the summary query results to Excel or PDF

Viewing of detail records and list management and use will be discussed in later sections.
The query tree and form attributes are defined in a metadata file using JavaScript Object Notation (JSON). JSON is an emerging standard used to configure JavaScript web components, and used as the data interchange format between components. The metadata file is input into a JavaScript query tree engine, which generates the tree view in the UI. When a property is chosen from the query tree, the node is passed to a JavaScript component to generate the form input field.

Form processing
When the form is submitted, an AJAX call is made to a Struts Action sending the form parameters entered by the user. The Struts Action class generates a complex query criteria object that is passed to the DAO layer to construct the query. The complex query constructor generates a Hibernate HQL query that submits a SQL query to the Oracle database.

Results processing
The results of the query are packaged into a JSON object and passed back the UI. The results are then displayed using a grid control.
3.6.5.2 Presentation Layer

Technologies
For the presentation layer we use a mix of JSP, JavaScript, Ajax and metadata files to present the query interface to users. By using metadata files and generic code, we are able to present alternative versions of the query interface, without the need for additional coding. We have chosen the EXT-JS JavaScript libraries, to build the query tree shown in the Advance Search screens, and to build the summary table grids.

Query UI
The property query tree on the left panel of the page is built using a combination of JavaScript and metadata file, which represents the entities and the properties that can be queried by the user. As the users clicks on query properties from the tree, an input field is displayed in the right side panel. The input field can be a simple input text box, drop down list for controlled vocabulary terms, or the user can click on a link to display a popup that lists all terms previously entered for this property.

Query Results
The results of the query are displayed in a tabular grid, using EXT-JS components. The user has several options:

1. Sort results based on column selection.
2. Selecting rows, for detail view.
3. Selecting rows to save in a result set.
4. Export the results to a file, that can be downloaded

3.6.5.3 Form Processing, Business Object Layer
Because the number of different properties that can be chosen by the user to qualify the query is subject to change, we have developed a generic methodology for processing the parameters and constructing the SQL to be submitted. In the Action class, based on the parameters supplied, a ComplexSearchCriteria will be constructed. The ComplexSearchCriteria object has properties to contain the project Id’s and the Name of the primary object of interest (Subject, Experiment, etc). In addition, it contains a collection of items that represents the properties and values passed in by the form submission. Each item will contain the value entered for this property, the name of the property and the search operator used to qualify the join to this property. The field name contains information representing the object and the property of the object that is being queried. A query operator can be “Exact Match”, “Like”, “Greater Than”, etc. For example, if the user is looking for Subjects that are “Male” and have Biological_Samples where the sample_type_name is “Tissue”, the action would be passed the following parameters:

StartingObject: Soi
SoiGenderValue: Male
SoiGenderOperator: “Exact Match”
BsSampleTypeNameValue: “Tissue”
BsSampleTypeNameOperator: “Exact Match”

3.6.5.4 Query Construction

The query constructor engine uses metadata that represents the paths between objects and information from the Hibernate mapping files to construct a query similar to the one in the section below. The Hibernate criteria classes are utilized to dynamically build the Hibernate Query language where clauses based on the property, value, and operator passed from the UI through form submission. The names of the properties in the UI tier are named to match the object properties in the Hibernate mapping files to simplify this construction.

Hibernate Query
An example of the resulting Hibernate query is show below.

```sql
select distinct soi.id
from SubjectOrgInfo soi
inner join soi.biologicalSampleSet bs
where soi.researchProjectInfo.id in (:projectIds)
  and soi.gender = 'Male'
  and bs.bsType = 'Tissue'
```

SQL Query
An example of the resulting SQL query generated by Hibernate from the Hibernate query is shown below:

```sql
select * from
  ( select distinct subjectorg0_.SUBJECT_ORG_ACC_NUM as col_0_0_
  from SUBJECT_ORG_INFO subjectorg0_
  inner join BIOLOGICAL_SAMPLE biological1_ on
    subjectorg0_.SUBJECT_ORG_ACC_NUM=biological1_.SUBJECT_ORG_ACC_NUM
  where (subjectorg0_.PROJECT_ID in (?))
    and (subjectorg0_.GENDER) = 'Male'
    and (biological1_.SAMPLE_TYPE_NAME = 'Tissue') )
where rownum <= ?
```

3.6.5.5 Query Processing – Results Display

The SQL query generated by query constructor is then submitted to the Oracle database. The results are processed and packaged into a JSON object. The JSON object is then passed back to the AJAX call that initiated the form submission, and displayed using the Table Grid control.

3.6.5.6 Search Class Diagrams

Action – Presentation Layer
EJB

Diagram showing class relationships in EJB architecture.
3.6.5.7 Research Data Detail Page Design Packages

The detail screens for the research entities, with the exception of the study detail screen, all follow a similar pattern and use the same technology stack. The top panel on the detail screen shows the detailed information for the entity. Below this panel there are a series of panels that can be expanded or collapsed and are loaded on demand. These panels re-use the same technology used to display the summary grids in the advanced search module described previously.

Many of the panels in the study detail screen represent aggregated information and cannot be easily satisfied with a simple SQL query. Therefore many of these panels require the aggregation of results to construct in the business layer.

The user may request to view one or more detail pages for a specific entity. For this discussion we will assume the user requested to see the details for 2 subjects: SUB1 and SUB2. This request starts by returning a screen containing “tabs” across the top, where tab 1 one represents SUB1 and tab 2 represents SUB2. As this page is displayed an AJAX call is submitted to an Action
class requesting the detailed information for the first tab. The action class calls an ejb class, which calls a dao class. The results are then merged with the JSP page, and returned to the AJAX call as HTML. This HTML is then placed in the DOM and displayed. The remainder of the page displays collapsed panels that represent entities that can be associated to SUB1. The contents of these panels are not available at this time. To retrieve the content for a panel the user clicks on the expand button, which initiates an AJAX call to retrieve a summary grid representing the associated entities. This AJAX call uses the same technology as outlined in the advanced search portion of this document, to return the summary grid.

If the user clicks on tab 2, then the whole process described above repeats. If the user then clicks on tab 1 again, results are shown immediately since this content was cached in the DOM.

The design package that follows utilizes the subject object detail page as an illustrative example of how all of the research data detail pages are implemented given that the same search and UI framework is used in all pages. The final section summarizes the detail page entities and the panels that are implemented in each one.
**Class Diagram – Subject Detail**

This diagram for subject detail is representative for how all the other research detail screens are constructed.
Class Diagram – Subject Panels

The classes used to construct the panel grids are the same as the classes used to construct the advanced search grids.
## Table of Entities and Panels

<table>
<thead>
<tr>
<th>Entity</th>
<th>Panel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subject</td>
<td>Subject Detail</td>
</tr>
<tr>
<td></td>
<td>Treatments</td>
</tr>
<tr>
<td></td>
<td>Protocols</td>
</tr>
<tr>
<td></td>
<td>Biological Samples</td>
</tr>
<tr>
<td></td>
<td>Assessments</td>
</tr>
<tr>
<td></td>
<td>Lab Tests</td>
</tr>
<tr>
<td></td>
<td>Adverse Events</td>
</tr>
<tr>
<td></td>
<td>Concomitant Medications</td>
</tr>
<tr>
<td>Biological Sample</td>
<td>Biological Sample Detail</td>
</tr>
<tr>
<td></td>
<td>Treatments</td>
</tr>
<tr>
<td></td>
<td>Protocols</td>
</tr>
<tr>
<td></td>
<td>Associated Experiments</td>
</tr>
<tr>
<td></td>
<td>Biological samples derived from this Biological Sample</td>
</tr>
<tr>
<td></td>
<td>Clinical Lab Tests</td>
</tr>
<tr>
<td>Experiment Sample</td>
<td>Experiment Sample Detail</td>
</tr>
<tr>
<td></td>
<td>Result Files</td>
</tr>
<tr>
<td></td>
<td>Protocols</td>
</tr>
<tr>
<td></td>
<td>Biological Samples</td>
</tr>
<tr>
<td></td>
<td>Reagents</td>
</tr>
<tr>
<td>Experiment</td>
<td>Experiment Detail</td>
</tr>
<tr>
<td></td>
<td>Experiment Samples associated with this Experiment</td>
</tr>
<tr>
<td></td>
<td>Protocols</td>
</tr>
<tr>
<td>Protocol</td>
<td>Protocol Detail</td>
</tr>
<tr>
<td></td>
<td>Experiments</td>
</tr>
<tr>
<td></td>
<td>BioSamples</td>
</tr>
<tr>
<td></td>
<td>Experiment Samples</td>
</tr>
<tr>
<td></td>
<td>Subjects</td>
</tr>
<tr>
<td></td>
<td>Studies</td>
</tr>
<tr>
<td>Reagent</td>
<td>Reagent Detail</td>
</tr>
<tr>
<td></td>
<td>HLA Typing System</td>
</tr>
<tr>
<td></td>
<td>Analytes</td>
</tr>
<tr>
<td></td>
<td>Reagent Sets</td>
</tr>
<tr>
<td>Study</td>
<td>Study Detail</td>
</tr>
<tr>
<td></td>
<td>Inclusion Exclusion Criteria</td>
</tr>
<tr>
<td></td>
<td>Demographics</td>
</tr>
<tr>
<td></td>
<td>Assessments</td>
</tr>
<tr>
<td></td>
<td>Concomitant Medications</td>
</tr>
<tr>
<td></td>
<td>Treatments</td>
</tr>
<tr>
<td></td>
<td>Adverse Events</td>
</tr>
<tr>
<td></td>
<td>Clinical Lab Tests</td>
</tr>
<tr>
<td></td>
<td>Mechanistic Assays</td>
</tr>
<tr>
<td>Documentation</td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td></td>
</tr>
<tr>
<td>Glossary</td>
<td></td>
</tr>
</tbody>
</table>
3.6.5.8  **Research Data Management Design Packages**

Based on the users permissions they may have the ability to edit the research entities by either updating the primary information of an entity or by creating or removing a relationship to another entity. Because editing of all research entities works in a similar fashion, we will use the editing of the subject entity to illustrate the process.

**Displaying the Edit Screen**

From the subject detail page, the “Edit Subject Information” link is displayed.

![Subject Edit Screen](image)

After choosing to update a subject the edit subject detail screen will be displayed. In the top section of the screen the previously entered values will appear in input text boxes or in drop down lists. The user makes the necessary changes and then clicks the Save Changes button.
Display Edit Screen Class Diagram

```
DisplayUpdateSubjectAction
  + execute()

SubjectManager
  + getEvents()
  + getAssociatedTreatments()
  + getAssociatedProtocols()
  + getAssociatedBioSamples()

DisplayAssociatedSubjectEntitiesAction
  + execute()

SubjectDAO
  + getEvents()
  + getAssociatedProtocols()
  + getAssociatedTreatments()
  + getAssociatedBioSamples()
```

Fields marked with an asterisk * are required.
Display Edit Screen Sequence Diagram
3.6.5.9 **Update Subject**
After the save changes button is clicked, the transaction is sent to the database to update the appropriate fields.

**Update Subject Class Diagram**

**Update Subject Sequence Diagram**
Delete Subject
Additionally, if the user has appropriate permissions he may delete the subject from the detail page.

Delete Subject Class Diagram

Delete Subject Sequence Diagram
Add Treatment

The next section shows any treatments currently linked to this subject. In ImmPort treatments are not considered primary research entities, but are considered as children of subjects. In this section you can add a new treatment, delete a treatment or edit an existing treatment.

Add Treatment Class Diagram

Add Treatment Sequence Diagram
From the summary grid for treatments in the main subject update page, the delete treatment button allows the user to remove a treatment from the database.

**Delete Treatment Class Diagram**

**Delete Treatment Sequence Diagram**
3.6.5.11 **Update Treatment**

Treatments may also be updated in the same interface as was shown earlier for entering a new treatment with the prior values now populated.

**Update Treatment Class Diagram**

![Update Treatment Class Diagram](image)

**Update Treatment Sequence Diagram**

![Update Treatment Sequence Diagram](image)
3.6.5.12 **Display Non-Associated Entities**

A subject record may be associated with many protocols and many biological samples.

To add additional associations, the user clicks the Add button in the appropriate grid and a window is displayed with a grid containing the available protocols or biological samples that may be associated to the subject. The sequence diagram below will only show how the non-associated biological samples are retrieved and displayed.
Display Non-Associated Entities Class Diagram
Display Non-Associated Entities Biological Sample Sequence Diagram
3.6.5.13  **Associate Entities**
After the user has chosen biological sample or protocols to link to the subject they submit the request using the following process.

**Associate Entities Class Diagram**

![Class Diagram](image)

**Associate Entities Sequence Diagram**

![Sequence Diagram](image)

3.6.5.14  **Disassociate Entities**
On the edit summary page the user may choose biological sample or protocols displayed in the associated entities grid to mark them for unlinking from the subject. The process is outlined below.
Disassociate Entities Class Diagram

- **Process: Disassociate Entity**: + execute()
- **SubjectManager**: + disassociateProtocol() + disassociateBioSample()
- **SubjectDAO**: + disassociateProtocol() + disassociateBioSample()

Disassociate Entities Sequence Diagram
Edit Subject Class Diagram