Originator

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Job Title | Signature | Date |
| Lloyd Kurth | Senior Software Architect |  |  |
| Brian Cullinan | Software Developer |  |  |

Approvals

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Job Title | Signature | Date |
| Scott Pletzer | Director of Quality & Program Management |  |  |
| Andrew Mason | Chief Technology Officer |  |  |

Revision History

| Revision | Revision Date | Reason for Revision/Change Request | Revised By |
| --- | --- | --- | --- |
| 000 | 18-Sep-12 | Initial Release | L. Kurth B. Cullinan |

Table of Contents

[1 Purpose 6](#_Toc335992227)

[2 Scope 6](#_Toc335992228)

[3 Definitions 6](#_Toc335992229)

[3.1 References 6](#_Toc335992230)

[3.2 Terms 6](#_Toc335992231)

[4 System Architecture 8](#_Toc335992232)

[4.1 Network 9](#_Toc335992233)

[4.2 Web Server 10](#_Toc335992234)

[4.3 Database Server 11](#_Toc335992235)

[4.4 Administration and Configuration 12](#_Toc335992236)

[4.5 ClearView Cloud 12](#_Toc335992237)

[4.6 ClearView Earth 13](#_Toc335992238)

[4.7 Purchase 13](#_Toc335992239)

[4.8 Support 13](#_Toc335992240)

[4.9 Web Services 13](#_Toc335992241)

[5 System Design 14](#_Toc335992242)

[5.1 Core 14](#_Toc335992243)

[5.2 Data Layer 14](#_Toc335992244)

[5.3 Third-Party Packages 14](#_Toc335992245)

[5.4 Languages 15](#_Toc335992246)

[5.5 User Interface 15](#_Toc335992247)

[5.6 Access Restrictions 15](#_Toc335992248)

[5.7 Roles 15](#_Toc335992249)

[5.8 Internationalization & Localization 16](#_Toc335992250)

[6 Component Design 17](#_Toc335992251)

[6.1 Shared Components 17](#_Toc335992252)

[6.1.1 Attributes 17](#_Toc335992253)

[6.1.1.1 Purpose 17](#_Toc335992254)

[6.1.1.2 Attributes 18](#_Toc335992255)

[6.1.2 Validation 20](#_Toc335992256)

[6.1.3 DataTables 20](#_Toc335992257)

[6.1.3.1 Purpose 20](#_Toc335992258)

[6.1.3.2 Methodology 20](#_Toc335992259)

[6.1.4 Widgets 22](#_Toc335992260)

[6.1.4.1 Purpose 22](#_Toc335992261)

[6.1.4.2 Widgets 22](#_Toc335992262)

[6.1.5 Menu 24](#_Toc335992263)

[6.1.5.1 Purpose 24](#_Toc335992264)

[6.2 Administration and Configuration 24](#_Toc335992265)

[6.2.1 Organization 24](#_Toc335992266)

[6.2.1.1 Purpose 24](#_Toc335992267)

[6.2.1.2 Function 25](#_Toc335992268)

[6.2.1.3 Data 28](#_Toc335992269)

[6.2.2 Location 28](#_Toc335992270)

[6.2.2.1 Purpose 28](#_Toc335992271)

[6.2.2.2 Function 29](#_Toc335992272)

[6.2.2.3 Data 35](#_Toc335992273)

[6.2.3 User 35](#_Toc335992274)

[6.2.3.1 Purpose 36](#_Toc335992275)

[6.2.3.2 Function 36](#_Toc335992276)

[6.2.3.3 Data 45](#_Toc335992277)

[6.2.4 Account 46](#_Toc335992278)

[6.2.4.1 Purpose 46](#_Toc335992279)

[6.2.4.2 Function 46](#_Toc335992280)

[6.2.5 Device 56](#_Toc335992281)

[6.2.5.1 Purpose 56](#_Toc335992282)

[6.2.5.2 Function 57](#_Toc335992283)

[6.2.5.3 Data 60](#_Toc335992284)

[6.2.6 Message 61](#_Toc335992285)

[6.2.6.1 Purpose 61](#_Toc335992286)

[6.2.6.2 Functions 61](#_Toc335992287)

[6.2.7 System Settings 63](#_Toc335992288)

[6.2.7.1 Purpose 63](#_Toc335992289)

[6.2.7.2 Functions 63](#_Toc335992290)

[6.3 ClearView Cloud 65](#_Toc335992291)

[6.3.1 Purpose 65](#_Toc335992292)

[6.3.2 Functions 65](#_Toc335992293)

[6.3.2.1 Patient Search 66](#_Toc335992294)

[6.3.2.2 Viewing Patient Details 66](#_Toc335992295)

[6.3.2.3 Listing Patient-Treatments 67](#_Toc335992296)

[6.3.2.4 Treatment Search 67](#_Toc335992297)

[6.3.2.5 Viewing a Treatment 68](#_Toc335992298)

[6.4 ClearView Earth 70](#_Toc335992299)

[6.4.1 Purpose 71](#_Toc335992300)

[6.4.2 Functions 71](#_Toc335992301)

[6.4.2.1 Statistics 71](#_Toc335992302)

[6.4.2.2 Map 71](#_Toc335992303)

[6.5 Purchases 73](#_Toc335992304)

[6.5.1 Purchase History 73](#_Toc335992305)

[6.5.1.1 Purpose 73](#_Toc335992306)

[6.5.1.2 Functions 73](#_Toc335992307)

[6.5.2 Making a New Purchase 74](#_Toc335992308)

[6.5.2.1 Purpose 74](#_Toc335992309)

[6.5.2.2 Functions 74](#_Toc335992310)

[6.5.3 Managing Credit Cards 76](#_Toc335992311)

[6.5.3.1 Purpose 76](#_Toc335992312)

[6.5.3.2 Functions 77](#_Toc335992313)

[6.5.4 Transferring Scans 78](#_Toc335992314)

[6.5.4.1 Purpose 78](#_Toc335992315)

[6.5.4.2 Functions 78](#_Toc335992316)

[6.6 Support Center 79](#_Toc335992317)

[6.6.1 Purpose 79](#_Toc335992318)

[6.6.2 Messages 79](#_Toc335992319)

[6.6.3 Composing 81](#_Toc335992320)

[6.7 Web Services 81](#_Toc335992321)

[6.7.1 Authentication 83](#_Toc335992322)

[6.7.1.1 Purpose 83](#_Toc335992323)

[6.7.1.2 Function 83](#_Toc335992324)

[6.7.1.3 Inputs 84](#_Toc335992325)

[6.7.1.4 Outputs 84](#_Toc335992326)

[6.7.1.5 Errors 84](#_Toc335992327)

[6.7.2 Registration 84](#_Toc335992328)

[6.7.2.1 Method: GetLocations 84](#_Toc335992329)

[6.7.2.2 Method: RegisterDevice 85](#_Toc335992330)

[6.7.2.3 Method: ResetAuthenticationKey 86](#_Toc335992331)

[6.7.3 Events 87](#_Toc335992332)

[6.7.3.1 Method: Create 87](#_Toc335992333)

[6.7.4 Scan Authorization and Recording 88](#_Toc335992334)

[6.7.4.1 Method: Validate 88](#_Toc335992335)

[6.7.4.2 Method: GetCount 89](#_Toc335992336)

[6.7.4.3 Method: ScanComplete 89](#_Toc335992337)

[6.7.5 Data Synchronization 90](#_Toc335992338)

[6.7.5.1 Patient Service 90](#_Toc335992339)

[6.7.5.2 Image Service 91](#_Toc335992340)

[6.7.5.3 Calibration Service 92](#_Toc335992341)

[6.7.5.4 Treatment Service 93](#_Toc335992342)

[6.7.6 Messaging 95](#_Toc335992343)

[6.7.6.1 Method: GetCollection 95](#_Toc335992344)

[6.7.7 Exception Upload 96](#_Toc335992345)

[6.7.7.1 Method: Create 97](#_Toc335992346)

[6.8 Logging and System Diagnostics 97](#_Toc335992347)

[6.8.1 Action Logger 98](#_Toc335992348)

[6.8.2 Exceptions 98](#_Toc335992349)

[6.8.3 Logging and Log4Net 98](#_Toc335992350)

[6.8.4 Auditing 99](#_Toc335992351)

[6.9 Device State 99](#_Toc335992352)

[6.9.1 States and Transitions 99](#_Toc335992353)

[6.9.1.1 Into New State 100](#_Toc335992354)

[6.9.1.2 New to Active 100](#_Toc335992355)

[6.9.1.3 New to Retired 100](#_Toc335992356)

[6.9.1.4 Active to Transitioning 101](#_Toc335992357)

[6.9.1.5 Active to Locked 101](#_Toc335992358)

[6.9.1.6 Transitioning to Active 101](#_Toc335992359)

[6.9.1.7 Transitioning to Locked 101](#_Toc335992360)

[6.9.1.8 Transitioning to Retired 101](#_Toc335992361)

[6.9.1.9 Locked to Active 101](#_Toc335992362)

[6.9.1.10 Locked to Transitioning 101](#_Toc335992363)

[6.9.1.11 Locked to Retired 101](#_Toc335992364)

[6.9.1.12 Retired is Final 102](#_Toc335992365)

[6.9.2 Device Registration 102](#_Toc335992366)

[7 Requirements Traceability Matrix 103](#_Toc335992367)

# Purpose

The purpose of this document is to describe the architecture, design and implementation of EPIC Central.

# Scope

The scope of this document is the design of EPIC Central and the interfaces it provides to users and ClearView devices.

# Definitions

Throughout this specification document references are made to the following documents and terms.

## References

|  |  |
| --- | --- |
| SR-002.001 | EPIC Central Requirements |
| SS-004.001 | EPIC ClearView Synchronization Service Software Specification |
| [MVC](http://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller) | Model-View-Controller on Wikipedia |
| [REST](http://en.wikipedia.org/wiki/Representational_state_transfer) | Representational State Transfer on Wikipedia |
| [LLBLGen Pro](http://www.llblgen.com/) | The LLBLGen Pro web site |
| [Authorize.Net](http://developer.authorize.net/) | The Authorize.Net Developer Center |
| [jQuery](http://jquery.com/) | The jQuery home page |
| [Google Maps](https://developers.google.com/maps/) | The Google Maps developer site |
| [Google Translate](http://support.google.com/translate/) | The Google Translate help page |

## Terms

|  |  |
| --- | --- |
| Administrator | Either a Service Administrator or an Organization Administrator; see those definitions below. |
| AJAX | Asynchronous JavaScript and XML. A programming model that uses the asynchronous request capabilities of modern web browser to issue background requests to servers and manipulate the display using the results. Enables highly interactive and responsive user interfaces. |
| API | Application Programming Interface. An interface specification for access to a software service implementation. |
| Atomic Transaction | A related set of database operations that either all occur or nothing occurs. |
| Device | A device is a ClearView device, more specifically the ClearView scanner along with its host PC running the ClearView application. To be able to operate, a ClearView device must communicate with EPIC Central. A device must be configured with EPIC Central before it can become operational. |
| Ecosystem | The worldwide combination of the EPIC Central service and all managed ClearView devices. |
| Firewall | A network appliance that inspects information packets flowing between a trusted, secure network and a public, insecure network to determine whether or not to allow packets to pass through, thereby providing a high level of network security. Packets are allowed or denied based on user-defined rules. |
| HTTP | The predominant and ubiquitous Internet protocol for accessing remote servers via a web browser. |
| HTTPS | HTTP through an SSL/TLS tunnel. Allows strong encryption of all data transmission with server authentication while remaining fully transparent to web application and user. |
| jQuery | An open-source JavaScript library for client-side, in-browser scripting of HTML and related standards. |
| Load Balancer | A network appliance that receives inbound network requests and distributes them across multiple servers to achieve redundancy and better throughput. |
| Location | An organization’s place of business. A single organization may have multiple locations where it conducts business. |
| MD5 | A cryptographic hash function the computes a 128-bit digest value for a set of input bytes. Not as secure as the SHA-2 family of hash functions but is less compute intensive and an excellent choice for transient applications. |
| MVC | Model-View-Controller, a design pattern that separates data and business logic (the model) from presentation (the view) and user interaction (the controller). |
| Organization | A legal entity with which EPIC has a business relationship, either directly or indirectly, and whose personnel need the ability to use EPIC Central. |
| Organization Administrator | A pre-defined role assigned to users who have administrative rights for a particular organization. This term is used throughout this document to refer to any user in this role. |
| REST | REpresentational State Transfer is a software architectural style for distributed systems where a client makes a request to a server that processes the request to completion and provides a response. The server is stateless; the request must provide all input the server needs to process the request. |
| Role | Defines a set of privileges/permissions used to manage access to system functions. A user assigned a particular role has the privileges defined by that role. |
| Service Administrator | A pre-defined role assigned to user who have full administrative rights for the entire EPIC Central system. This term is used throughout this document to refer to any user in this role. |
| SHA-512 | A cryptographic hash function that computes a 512-bit digest value for a set of input data bytes. |
| Shared Secret | A piece of data known only to the parties involved in a secure communication. |
| SSL/TLS | Secure Sockets Layer/Transport Layer Security. Provides an encrypted tunnel over a public network through which many other protocols, such as HTTP and FTP, can operate transparently. TLS is the current specification and replaces SSL, but the SSL acronym is still widely used. |
| Treatment | The record of a scan. This term is used interchangeably with “scan” and is kept to maintain continuity with the terminology of the ClearView application. |
| User | A person that can log in to use EPIC Central. This person may be either an EPIC employee, or a person associated with another organization with a business relationship with EPIC, such as a customer using a ClearView device. |
| User Account | A record, maintained with the database, of a user. A user account contains a unique user name which identifies the account. It also contains other information about the user, e.g. first name, last name, email address, and information about the user’s activity. |
| Web Application | A highly-interactive, and often complex, application built with web standards (HTML, CSS, JavaScript, etc.) that runs on a web server and allows the use of a standard web browser as the user interface device. |
| Web Service | An application programming interface (API) implemented and running as a service on a server that can be invoked by a client via the network. |

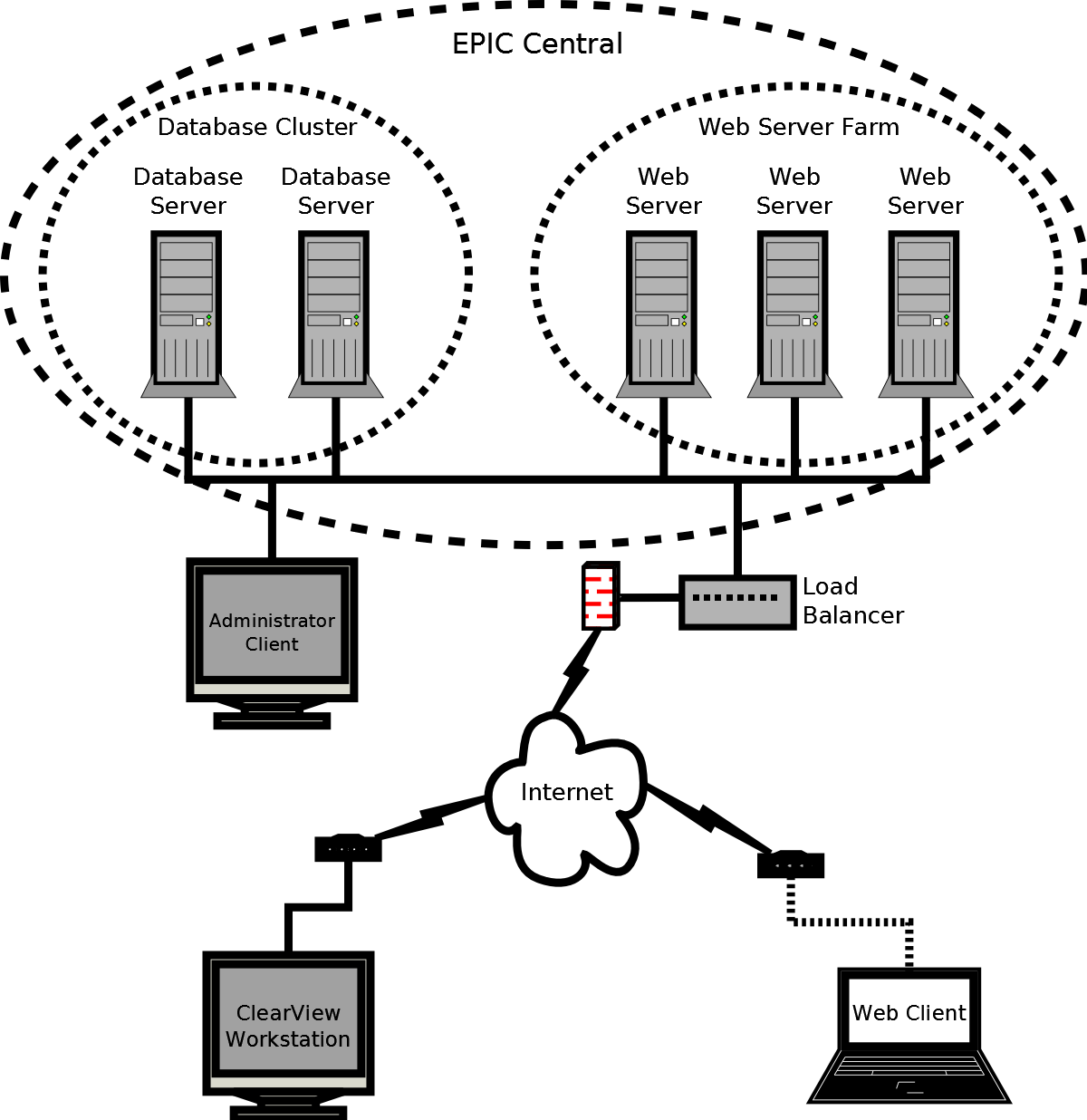
# System Architecture

This section provides a high-level view of the overall system architecture for EPIC Central. More detail is provided later in this specification.

## Network

At the highest level, EPIC Central is a web application running on a web server with a database for persistent storage of all application artifacts. Initially, the configuration will be a single web server with a single database server. As it grows, it will be expanded to multiple web servers and multiple database servers to provide greater throughput and redundancy.

The web servers will be connected to the Internet through a firewall and load balancer that will distribute the workload. Connections to the web application will be made from ClearView workstations and standard web clients running browsers. There will be one or more administrator clients connected to the internal network. The following diagram shows this basic configuration:



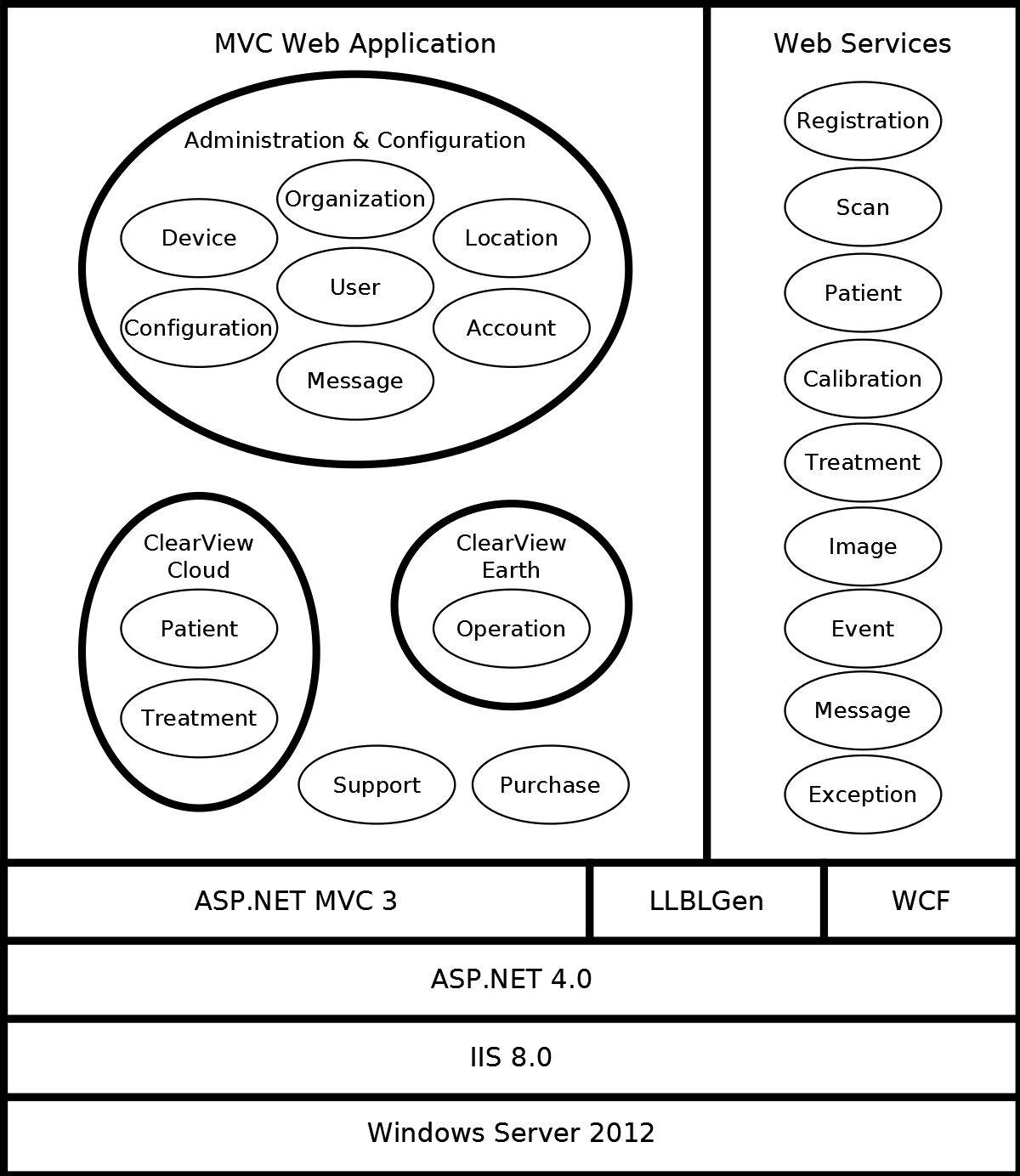
## Web Server

The web server hosts the core application and services that make up EPIC Central. It is an ASP.NET application running on Internet Information Services (IIS) 8.0 and Windows Server 2012. The current implementation is written to the ASP.NET 4.0 and ASP.NET MVC 3 APIs. There are newer versions of these APIs and the next release will likely use those.

The web services are lightweight REST compliant services. They use the Windows Communications Framework (WCF). The input and output of the web services is XML-based, but uses a minimalist approach rather than full standards compliant web services.

Both the web application and web service use the LLBLGen Pro v3.5 object-relational mapping package for accessing the database.

The following diagram shows the high-level design of the web server. The various components are discussed briefly below and in greater detail later in this specification.



## Database Server

The database server configuration is simple and standard. The database is SQL Server 2012 running on Windows Server 2012.

The full database schema contains dozens of tables. But a core set of only eight tables will provide a solid foundation for understanding this specification. The following database diagram shows the core schema:



Here is a brief explanation of the tables in this diagram:

* Organization – This table is the root of the schema. An entry represents an organization such as a customer, distributor, or the single host for EPIC Central.
* Location – An entry in this table represents a location, or place of business, for an organization. Every organization must have at least one location, but can have many.
* User – This table contains user accounts. Each user that needs to access EPIC Central must have a user account. Users are associated with an organization.
* UserAssignedLocation – This table joins a User to a Location. To have access to a location a non-administrator user must be assigned to the location by an administrator.
* Device – An entry in this table represents a ClearView device. Entries are added by a Service Administrator. A represented device must register to establish access to EPIC Central.
* Patient – An entry in this table represents a patient who has been scanned. These records are uploaded by devices.
* Calibration – An entry in this table represents the record of a calibration performed by a device. Entries are created when calibrations are uploaded by devices.
* Treatment – An entry in this table represents a treatment, or scan, that was performed on a patient. A treatment is also associated with a calibration, which indirectly associates it with a specific device. Treatment entries are created when treatment records are uploaded by devices.

All other database tables, except a couple for settings and status, are associated directly or indirectly with one of these core tables.

## Administration and Configuration

The Administration & Configuration package is a set of components for configuring and managing all the entities that determine the dynamic operations of the system. It consists of the following components:

* *Organization* – An MVC component that provides the ability for a Service Administrator to view, create and modify organizations.
* *Location* – An MVC component that provides the ability for both Service Administrators and Organization Administrators to view, create and modify locations, and for other users to view their organization.
* *User* – An MVC component that provides the ability for both Service Administrators and Organization Administrators to view, create and modify user accounts, and for other users to view other users assigned to their locations.
* *Account* – An MVC component that provides user account functions for any user, including log on, log off, registration and password management.
* *Device* – An MVC component that provides the ability for a Service Administrator to view, create and modify devices.
* *Message* – An MVC component that provides the ability for a Service Administrator to create and manage messages to be delivered to devices.
* *Configuration* – An MVC component that provides the ability for a Service Administrator to change system settings.

## ClearView Cloud

The ClearView Cloud package allows any user that has appropriate permissions to view patients and their scans through an interface similar to and consistent with the ClearView application’s user interface. This allows a user to access patient scans from anywhere on the Internet using a standard web browser. It consists of the following components:

* *Patient* – An MVC component that provides the ability to find and view patient information.
* *Treatment* – An MVC component that provides the ability to view a patients’ treatments (scans), including all images and diagnostic results.

## ClearView Earth

The ClearView Earth package allows Service Administrators to view current summary system metrics and events as they occur in near real-time. The view is a map of the world with icons showing the locations of ClearView devices. Each icon indicates the last event received from the device it represents and changes as different events occur.

*Operation* is an MVC component that implements the functionality for ClearView Earth, watching for events, updating the map and the metrics.

## Purchase

*Purchase* is an MVC component that allows users to purchase scans for any of their devices and to transfer scans from one device to another. A device must have scans available before it can perform a scan. Each time a scan is completed, the available scan count is decremented. When the count is depleted, more scans must be purchased.

## Support

*Support* is an MVC component that allows any user to send a request for support to a System Administrator or a message to any other user, and for the receiver to reply. It works very similarly to an email system.

## Web Services

A set of REST web services allows ClearView devices to communicate directly with EPIC Central in a fully automated way. The following services have been implemented:

* *Registration* – provides the ability to register a device with EPIC Central.
* *Scan* – provides the ability for a device to validate its own ability to perform a scan, get the number of scans it has available, and report the completion of a scan.
* *Patient* – provides the ability for a device to upload and update patient records.
* *Calibration* – provides the ability for a device to upload calibration metrics.
* *Treatment* – provides the ability for a device to upload all the diagnostics variables and calculations related to a treatment (scan).
* *Image* – provides the ability for a device to upload a set of images related to either a calibration or a treatment.
* *Event* – provides the ability for a device to report events as they occur.
* *Message* – provides the ability for a device to download any messages that an administrator has created to be sent to it.
* *Exception* – provides the ability for a device to report software exceptions so they can be viewed and diagnosed.

In addition to these specific services, the web services package implements a custom authentication mechanism so once registered a device can authenticate itself to access the web services. No unauthenticated access is allowed.

# System Design

This section provides an overview of the overall system-level design. A good understanding of these core design topics is essential to a deeper understanding of the various components presented later.

## Core

Keeping with EPIC’s standard choice of platform, EPIC Central is designed and implemented on the Microsoft stack. This includes the following components:

* Windows Server 2012 – the base operating system for both the web server and database server.
* SQL Server 2012 – the database software package.
* Internet Information Services (IIS) 8.0 – the web server package.
* ASP.NET 4.0 – the web application framework.
* ASP.NET MVC 3.0 – the model-view-controller framework for ASP.NET.
* Windows Communication Foundation – the distributed computing package for service-oriented architecture (SOA).

Server sessions are implemented using the database in the standard ASP.NET manner. This will allow easy migration to a farm of web servers in the future; all web server instances will be stateless.

## Data Layer

All data access uses the LLBLGen Pro object-relation (O/R) mapping tool. LLBLGen allows the application to access data objects as “entities,” each representing a row of data in a database table. LLBLGen handles the low-level access to SQL Server, including transactions, along with fetching related entities when necessary. For EPIC Central, the database schema was designed first and LLBLGen tools used to generate the O/R layer.

All data updates that modify multiple data tables or rows use appropriate transactions to ensure updates are atomic and either fully succeed or fail with no modifications made. *[Req: 4.6.8]*

## Third-Party Packages

Other significant third-party software packages include the following:

* Authorize.NET APIs – Authorize.NET is the credit card payment processor and this package is the one they provide for interfacing to their systems.
* AE.Net.Mail – an IMAP mail client package. This is used for interfacing to an email server in the *Support* component.
* Log4Net – an application logging package which supports logging levels and multiple output sinks.
* jQuery – the ubiquitous JavaScript client package.
* jQuery UI – the user interface component package for use on top of jQuery.
* jQuery Validation – the standard client-side validation package.
* DataTables – a jQuery grid implementation; used extensively to view data in a grid layout.
* Layout – a jQuery space layout package used for the base page layout.

## Languages

All server code for the EPIC Central application is written in C#. There is also a significant amount of JavaScript that is downloaded and run in the browser client. AJAX is used where appropriate for better overall user experience.

## User Interface

The user interface for EPIC Central uses standard HTML and CSS. It is designed to support the following web browsers:

* Microsoft Internet Explorer (IE), version 9.0 and above *[Req: 4.8.2.1]*
* Mozilla Firefox, version 11 and above *[Req: 4.8.2.2]*
* Google Chrome, version 18 and above *[Req: 4.8.2.3]*
* Apple Safari, version 5.1 and above *[Req: 4.8.2.4]*

The overall layout, color scheme and style for the user interface are designed to be consistent with the EPIC corporate website. *[Req: 4.8.5.1]*

Use of the standard jQuery UI package, along with CSS objects applied in a consistent manner, provide a consistent look-and-feel to the entire web application. *[Req: 4.8.5.3]*

## Access Restrictions

The public web interface will be restricted to HTTP through an SSL/TLS tunnel (HTTPS). This ensures that a web browser client will authenticate the EPIC Central server when connecting and that all data transferred between client and server, and vice versa, will be encrypted to prevent unauthorized access to sensitive data. *[Req: 4.8.1.1]*

Access to the administrative console for both the web server(s) and database server(s) will be restricted to the local network; no access from the public Internet is allowed. Furthermore, all Windows console access will be restricted to users who are members of the standard Windows Administrators group. *[Reqs: 4.8.1.2, 4.8.1.3]*

## Roles

Roles are predefined in the current release of EPIC Central. This may change in the future to allow more flexibility in assigning access permissions. There are currently three roles *[Req: 4.1.5.3.1]:*

* Service Administrator – a user in this role has access to the entire system.
* Organization Administrator – a user in this role has full access and administrative rights for his/her own organization.
* Simple User – an ordinary user with no administrative rights.

Each user has one, and only one, role. A user’s role must be set when the user’s account is added. It can be changed later by an administrator. If a user’s role is changed, the change takes effect on the next action performed by the user. A user in the Service Administrator role can assign any role to another user, or to himself/herself. A user in the Organization Administrator role can assign the Organization Administrator role or the Simple User role to any user in his/her organization, but cannot assign the Service Administrator role to anyone. A user in the Simple User role cannot assign roles. *[Reqs: 4.1.5.2, 4.1.5.4.1, 4.1.5.4.2, 4.1.5.4.3, 4.1.5.4.4, 4.1.5.4.5, 4.1.5.4.6, 4.1.5.4.7]*

A user in the Service Administrator role has access to all organizations and configuration, state and data of those organizations. This is the “superuser” role in EPIC Central. It can only be assigned to users associate with the service host organization, i.e. EPIC Central itself. *[Req: 4.1.5.3.2]*

A user in the Organization Administrator role has access to his/her own organization’s configuration and data, but no access to any other organization. This role can be assigned to any user. *[Req: 4.1.5.3.3]*

A user in the Simple User role has no administrative rights. Such a user has access only to his/her own settings and status and data for locations to which he/she is assigned. A user is assigned to locations when the user account is created. These assignments can be modified later by an administrator. *[Req: 4.1.5.3.4]*

A user’s role determined what functions a user can perform and what data a user can see. Access checks are implemented with attributes on methods and also with specific checks to see what role the logged in user has.

Through this document the term “a Service Administrator” refers to a user in the Service Administrator role. Likewise, the term “an Organization Administrator” refers to a user in the Organization Administrator role. And “a Simple User” refers to a user in the Simple User role. The term “an administrator” refers to a user in either the System Administrator role or the Organization Administrator role.

## Internationalization & Localization

Internationalization (I18N) is the process of creating an application in a way that enables it to be localized for a particular world region, locale and language. Localization (L10N) is the process of implementing a localized version for a region, locale and language. *[Req: 4.8.3.1]*

EPIC Central is designed for I18N through the use of standard ASP.NET resource files. All strings used for display in the user interface are contained in resource files. Each individual string has a name that can be used to reference its resource file definition. The C# code references strings by name enabling L10N simply by switching one resource file for another. Strings used in client-side scripts are loaded from the resource files when the page that uses the script is loaded or via an AJAX call to retrieve them.

There are multiple resources files. They are segregated based on where they are referenced and according to function. The names are specific for each use. This is necessary because the way a phrase is translated may differ based on the context where it is used.

L10N is achieved simply by creating the necessary resource files for a particular locale/language combination. Both a locale and language are required to create a unique combination. For example, US English is slightly different from UK English. L10N requires that all the strings in the resource files be translated to provide each L10N implementation.

The selection of locale and language is initially detected automatically based on the operating system settings of the user’s computer. Detection is performed using JavaScript running in the user’s browser. Each user can override this setting and select any available L10N. If the detected settings are not implemented by EPIC Central, the default will be US-English which is fully implemented. *[Reqs: 4.8.3.2, 4.8.3.3, 4.8.3.4]*

The EPIC Central project includes a resource translation application which can be executed to provide an initial translation for any L10N. This is intended to provide a starting point for a human translator to produce a deployable version. This application uses Google Translate.

# Component Design

The System Architecture section above provided a summary description of the system. This major section will provide design details for all the components identified by the architecture and some others that are shared.

## Shared Components

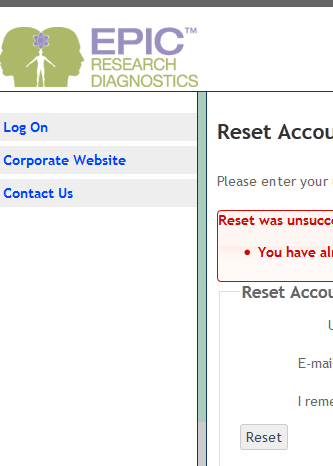
There are a number of components that are used and shared by other larger components these are defined first.

### Attributes

C# attributes are like meta-data, or “adjectives”, that describe other functions of the system. Some of these attributes contribute to important requirements within the system as a whole. These significant attributes are described below. However, this is not meant to be a complete list and it can be assumed that attributes not listed are implementation details beyond the scope of this document.

#### Purpose

The purpose of attributes is to provide a simple interface with potentially very complex implications. They can be used in various places but it all leads back to the same generalized functionality. A good example is the *Menu* attribute which is placed on a myriad of different functions to cause them to show up in the left side menu, most of which are described in this document; however, it is not applied to all functions, some of which are hidden from user’s sight.



As shown, the *[ActionMenu]* attribute is used on the *Log On* function, but the *Reset Account* page is currently being shown which has no *Menu* attribute. The *Menu* attribute is discussed in further detail in section 6.1.5.

#### Attributes

Below is a list of the most important attributes that affect the operations of EPIC Central as a whole.

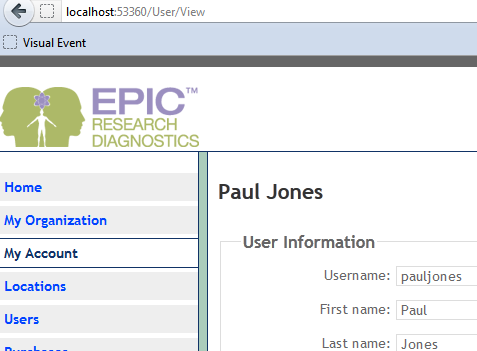
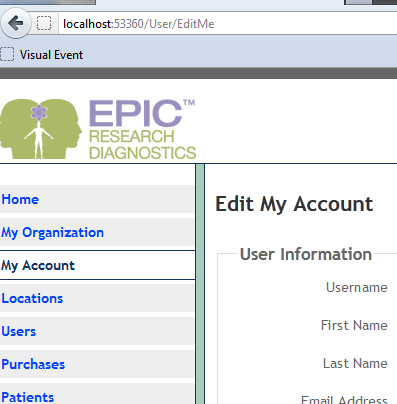
##### Menu

The *Menu* attribute is used as a marker for which functions should display on the left-hand menu. The most extensive menu is available to users in the Service Administrator role. They have access to the entire EPIC Central system.

The menu is generated by searching all functions for *Menu* attributes, then filtering the menu based on the currently logged in user’s access permissions.

The *Menu* attribute has a few important options:

1. *[ActionMenu(Rank = 400)]* can be used to specify the rank (in this case 400) of where it should appear relative to other *Menu* attributes throughout the system.
2. *[ActionMenu(ResourceName = “LocationList\_Locations”)]* can be used to specify the name of the resource that provides the translated text for the menu item, if it differs from the default assigned template of ControllerAction where Controller is the name of the controller and action is the method to which the attribute is assigned.
3. *[ActionMenu(SelectAction = “List”)]* can be used to indicate that another action, different from the action it is assigned to, should be highlighted when this action is accessed. An example of this is displayed below, when the user is editing their account, the *Menu* attribute for displaying their account (in non-edit mode) is selected:  
     
   Clicking *My Account* would lead back to the address “/User/View”



The menu for each role is displayed below. The Simple User menu is very similar to the Organization Administrator, so the delegation of access restrictions is the responsibility of each individual page’s content, rather than the menu.

|  |  |  |
| --- | --- | --- |
| Service Administrator | Organization Administrator | Simple User |
|  |  |  |

*[Reqs: 4.1.5.1, 4.8.5.4]*

##### Allow/Deny

The *Allow* and *Deny* attributes are an important function of the authentication and access rights system.

The *Allow* attribute can specify which users or roles are allowed to access a particular part of EPIC Central. The *Allow* attributes takes the following parameters:

* *Users* – a comma separated list of usernames to allow access to a particular function.
* *Roles* – a comma separated list of roles to allow.

The *Deny* attribute takes the same parameters, but reverses the result of the *Allow* attribute. This can be used to deny access to a particular user or role.

*[Reqs: 4.1.5.1, 4.8.5.4]*

##### Rate Limit

The *RateLimit* attribute limits the frequency at which a user can access a particular function of the system. This is used for security purposes. The *RateLimit* attribute is applied to functions such as resetting a password. If the user tries to perform the function multiple times in quick succession, the request will be blocked.

The *RateLimit* attribute takes the parameter *Seconds*, which is the number of seconds between allowed requests. If *Seconds* is not specified, the rate is initially set to 2 seconds and is multiplied by two with each request. If a user clicks the *Reset* button to reset their account 3 times, they will be blocked for 8 seconds after the third attempt.

### Validation

Validation is performed on all necessary inputs to the system.

### DataTables

*DataTables* is a jQuery plugin that is used throughout the system. *DataTables* is a common display that allows lists to be shown with features such as column headings that can be clicked to sort. *DataTables* is like the window into the EPIC Central database.

#### Purpose

*DataTables* allows data to be displayed throughout the site in a consistent manner. *DataTables* adds expected functionality such as clicking a column heading to sort, searching the data displayed, paging data to minimize page length and scrolling. *[Req: 4.8.5.3]*

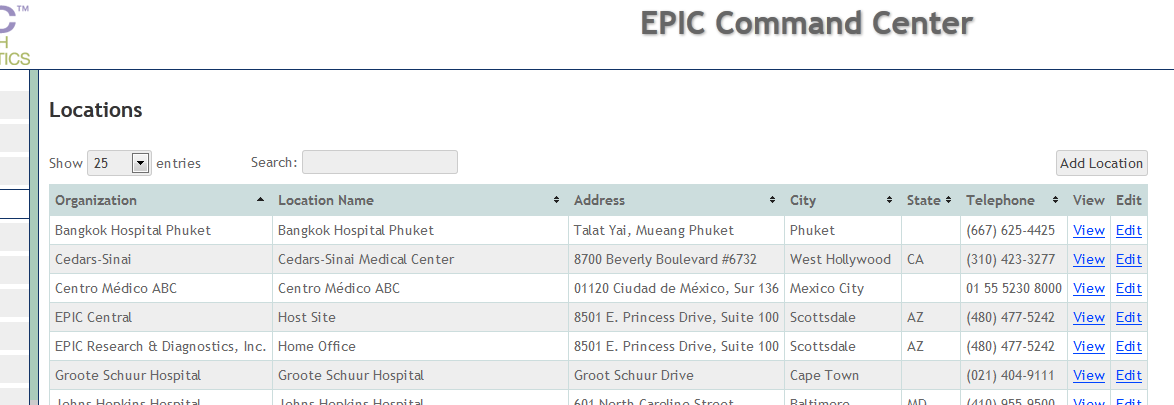
#### Methodology

There are a few main components that makeup the *DataTables* system. The client browser loads the jQuery plugin automatically for any page where a table is configured. The plugin then makes a call back to the server in order to populate the table with the necessary data. During this callback, the table configuration is reloaded and the parameters of what should be displayed are passed to the *DataTablesController.Query()* function. That function performs all of the business logic behind the scenes, loading all of the data with the proper bounds and ranges applied from the database, then returning the rendered text to the client browser.

To summarize this process:

This process allows *DataTables* to be very tightly controlled. Only the information specified in the configuration will ever be output to the client. Specialized functions, such as searching, can be applied on the server as described in the Algorithms section below.

Every component of a table is customizable. For some columns it doesn’t make sense to allow sorting because the column is the same in every row. For example, the *View* and *Edit* columns on the *Locations* page:



The table headings can be defined and translated. A column can be hidden when the text is combined with another column such as the *Device* column on the *Purchases* page:



Device state and the device serial number are combined in the Device column. The Device column is displayed. The device state and serial number are held in hidden columns in the table. Both of these hidden columns can be searched.

The displayed text can be formatted as described above, even though the data is stored differently than how it is displayed to the user.

### Widgets

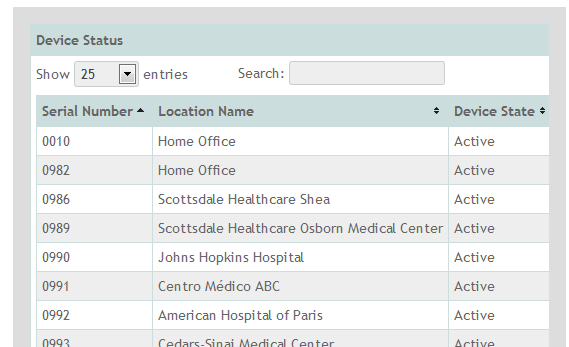
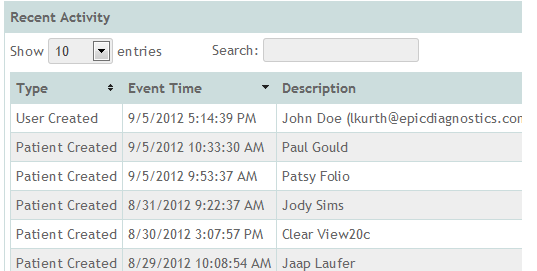
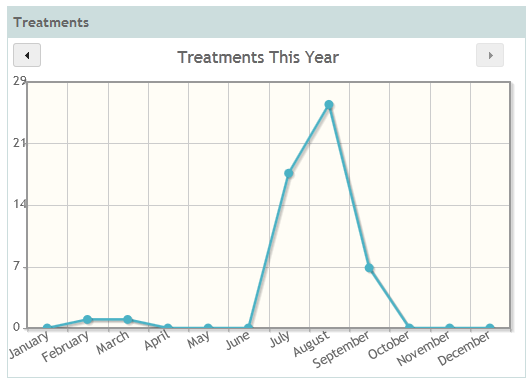
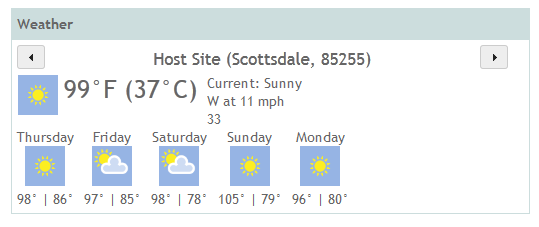
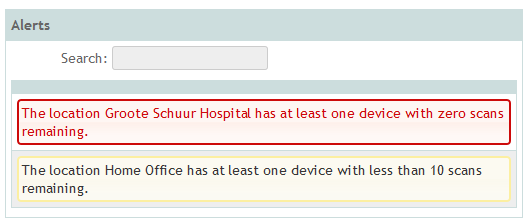
Widgets are an extension of the WidgetController that are displayed in drag-able boxes on the home page.

#### Purpose

Widgets are intended to be useful summaries of information that provide links to relevant and interesting data.

#### Widgets

There are various widgets that display different information:

1. Device State – Displays a list of the devices accessible by the current user, including each device’s serial number, location and state. The following image shows an example:  
   
2. Recent Activity – Displays a list of the 5 most recent treatments, patients, exceptions and/or users created within the EPIC Central system. The following image shows an example:  
   
3. Treatments – Displays a graph of how many treatments were performed in each month of a year. The year can be changed using the arrow controls. The following image shows an example:  
   
4. Weather – Shows the weather conditions for all the locations the current user has access to. The location can be changed using the arrow controls. The following image shows an example:  
   
5. Alerts – Shows alerts relevant to the logged in user. There are four different levels of alerts: *Error*, *Warning*, *Notice*, and *Default*. *Error* and *Warning* are shown in the following example:  
   

### Menu

The left-hand menu is the primary navigation for EPIC Central.

#### Purpose

The purpose of the menu is to provide the necessary links for users to navigate the system.

## Administration and Configuration

The components described in this section are for configuration and management of the core entities that dynamically make up the EPIC Central system.

### Organization

An MVC component that provides the ability for a Service Administrator to view, create and modify organizations, an Organization Administrator to view or modify his/her own organization, and any other user to view his/her organization.

Organization is a root-level data object to which are attached users and locations. Thus, no users or locations can exist without an organization.

#### Purpose

The Organization MVC component exists to provide the ability for users to view and administer organizations.

A Service Administrator is able to view the organizations currently defined and make modifications to them. A Service Administrator is also able to add new organizations as necessary. This functionality is restricted to users in the Service Administrator role. *[Reqs: 4.1.1.4, 4.1.1.5]*

Each organization has a name and a type. Both of these are required fields. The type is implemented as a selection list and contains all defined organization types from which the user can select when adding or modifying an organization. The initial release will support only two types, a *Host* type and an *End-User* type. These types are defined as an enumeration in the code. New types may be added in the future and the selection list will automatically include them. For example, there may be a “distributor” type to represent an organization that sells and distributes ClearView devices. *[Reqs: 4.1.1.1, 4.1.1.2]*

Only one organization will be allowed to be of type *Host*. This organization will represent EPIC as the EPIC Central service provider. It will be created at the time the database is created. When other organizations are added, they will always be *End-User* organizations. *[Req: 4.1.1.3]*

When an organization is added, a unique identifier for the organization will be generated. This ID is guaranteed to be unique across the ecosystem. *[Req: 4.7.2.1]*

A Service Administrator can change the name of an organization and make it active or inactive. If an organization is made inactive, it is effectively disabled. No user associated with the organization is allowed to log in and no device associated with the organization is allowed to authenticate. *[Reqs: 4.1.1.5, 4.7.3.2]*

An Organization Administrator can view his/her own organization and change the name. Any non-administrator user can view the organization but is not allowed to make changes. *[Req: 4.1.1.6]*

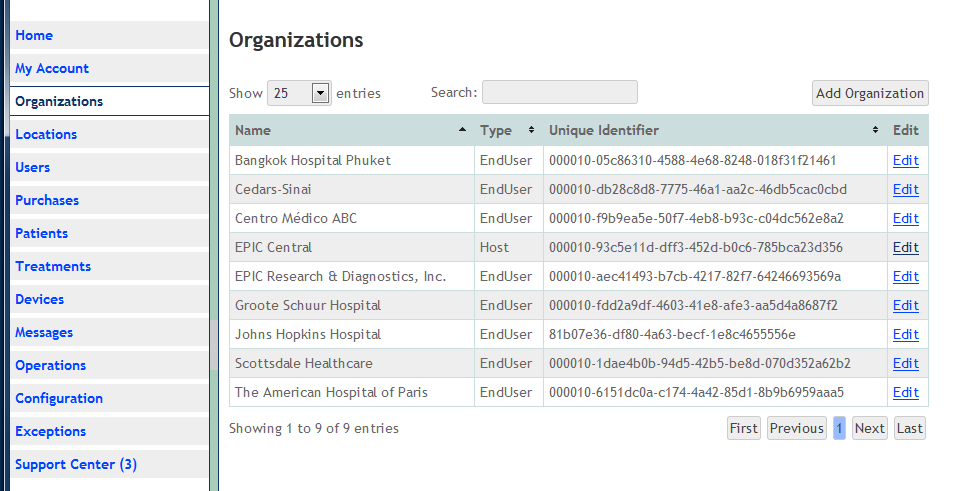
#### Function

This component provides all the necessary functionality to view and administer organizations. These functions include the following:

##### List of Existing Organizations

A Service Administrator is allowed to view a list of all the existing organizations. This functionality is available only to Service Administrators.

When a Service Administrator clicks the *Organizations* entry in the main menu, a list of the currently configured organizations will be presented:

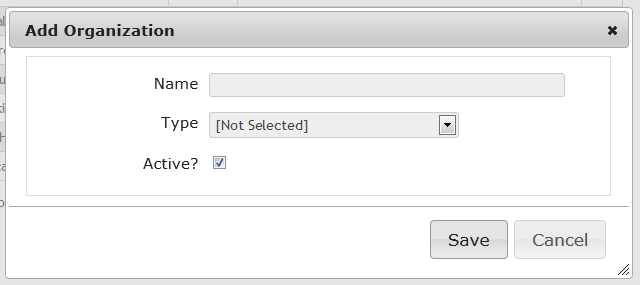


The list contains all information for each organization, including its name, type and unique identifier. There is no additional “view” function necessary to allow a Service Administrator to view organization details.

##### Add a New Organization

A Service Administrator is allowed to add a new organization. This functionality is available only to Service Administrators.

To add a new organization, a Service Administrator clicks the *Add Organization* button presented by the table that lists the organizations. When the button is clicked, the following pop-up dialog is displayed:



The Service Administrator must provide the name of the organization and select the type. The only type available in the selection list is “End User.” No other type of organization can be created. Normally, an organization is active when it is created. The Service Administrator can create an inactive organization by unchecking the *Active* checkbox.

When all input is provided, clicking the *Save* button will cause the organization to be created and the dialog box removed. Clicking the *Cancel* button at any time will remove the dialog box and return to the list with no modifications made.

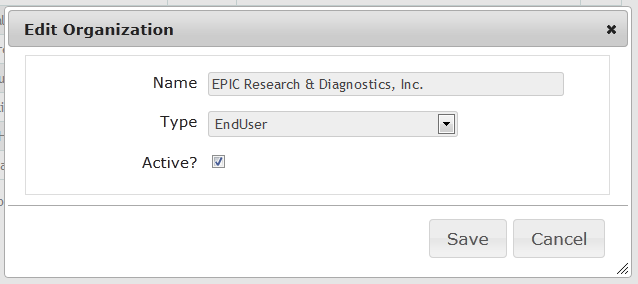
When the *Save* button is clicked, EPIC Central will validate that a name for the organization has been entered and a type selected. If either validation fails, an appropriate error message will be presented in the dialog box. The user can make corrections and resubmit or cancel.

When a new organization is created, EPIC Central will generate a unique identifier for it.

##### Edit an Existing Organization

A Service Administrator is allowed to edit an existing organization. This functionality is available only to Service Administrators.

To edit an organization, a Service Administrator clicks the *Edit* link in the rightmost column of the row that presents the organization to be edited. When the link is clicked, the following pop-up dialog is displayed:



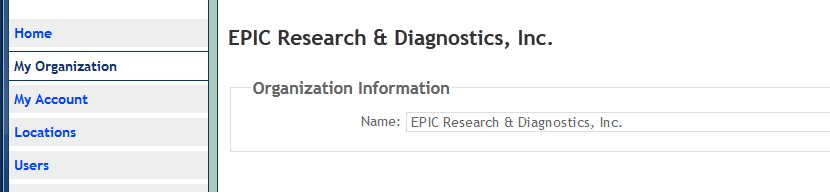
The Service Administrator can change the name of the organization or change the active state by checking or unchecking the *Active* checkbox. Since the only type allowed is “End User,” the type cannot be changed.

When all changes have been made, clicking the *Save* button will cause the organization to be updated and dialog box removed. Clicking the *Cancel* button at any time will remove the dialog box and return to the list with no modifications made.

When the *Save* button is clicked, EPIC Central will validate that a name for the organization has been entered and a type selected. If either validation fails, an appropriate error message will be presented in the dialog box. The user can make corrections and resubmit or cancel.

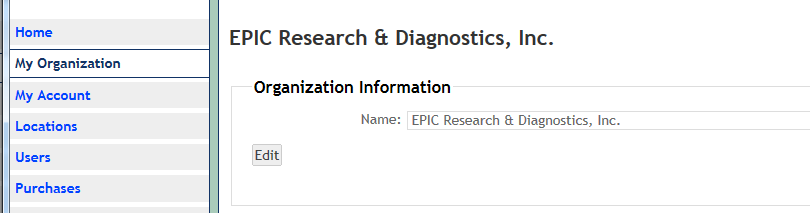
##### View My Organization

Users who are not Service Administrators are presented a *My Organization* entry in the main menu. When a Simple User chooses *My Organization*, the following is displayed:



This allows the user to view the name of his/her organization. No changes are allowed.

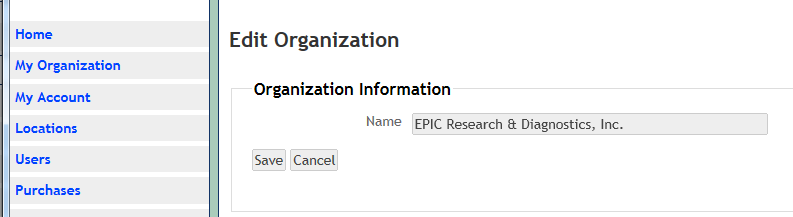
When an Organization Administrator chooses *My Organization*, the following is displayed:



The only difference is the presence of the *Edit* button that allows the Organization Administrator to change the name of the organization.

##### Edit My Organization

An Organization Administrator is allowed to change the name of his/her own organization. After choosing *My Organization* in the main menu, an Organization Administrator is presented with a view of the organization; see previous section. When the Organization Administrator clicks the *Edit* button, the view changes slightly:



The *Edit* button is replaced by *Save* and *Cancel* buttons and the *Name* field is now an editable textbox. The user can make changes and then click *Save* to save the changes, or click *Cancel* to return to the previous view.

#### Data

The Organization table in the database contains a record for each organization defined in EPIC Central. When a new organization is created, another record is added to the table. When an existing organization is modified, the existing record is updated.

No records are ever deleted from the Organization table. Making an organization inactive effectively removes it from the system. No user associated with the organization can log in and no device associated with the organization can authenticate. *[Req: 4.7.3.1]*

### Location

An MVC component that provides the ability for a Service Administrator to view, create and modify locations for any organization, an Organization Administrator to view, create and modify locations for his/her own organization, and any other user to view the locations to which he/she is assigned.

Location is also a data object and a table in the database. Devices are attached to locations as are patient records. Users are assigned to locations, which allows the user access to devices and other data associated with those locations.

#### Purpose

The Location MVC component exists to provide the ability for users to view and administer locations.

A Service Administrator is able to view all the locations currently defined for all organizations and make modifications to them. A Service Administrator is also able to add a new location for any organization. When adding a new location, or modifying an existing location, a Service Administrator is able to edit the geocoding information (latitude and longitude) for a location. Modification of the geocoding is restricted to users in the Service Administrator role. *[Reqs: 4.1.2.3, 4.1.2.5]*

An Organization Administrator is able to view all the locations currently defined for his/her own organization and make modifications to them. An Organization Administrator is also able to add a new location to that organization. *[Req: 4.1.2.4]*

Any non-administrator user can view the locations he/she is assigned to but is not allowed to modify any location.

Every location is required to have a name and should also have a mailing address and telephone number. These fields can be modified by either a Service Administrator or Organization Administrator. When a new location is added, the geocoding information for the address entered will be requested from an online service such as Google Maps. If it can be retrieved successfully, it will be saved with the location. Otherwise, a Service Administrator can enter it manually. *[Reqs: 4.1.2.1, 4.1.2.2, 4.1.2.6]*

When a location is added, a unique identifier for the location will be generated. This ID is guaranteed to be unique across the ecosystem. *[Req: 4.7.2.1]*

#### Function

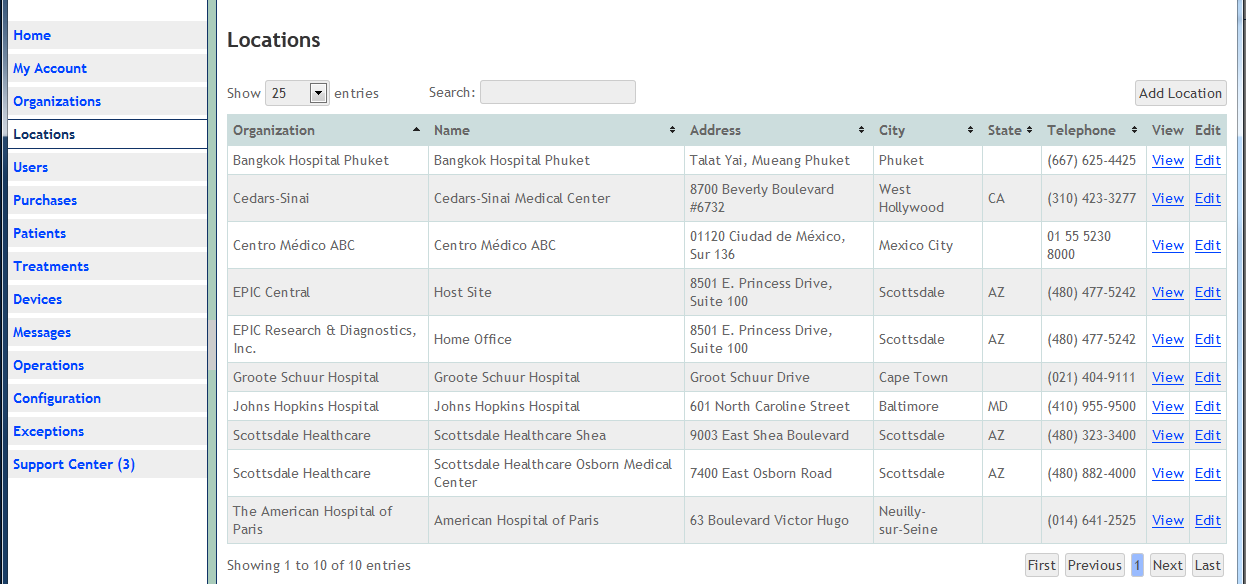
This component provides all the necessary functionality to view and administer locations. These functions include the following:

##### List of Existing Locations

The list of locations a user sees is dependent on the user’s role. A Service Administrator can see all the locations for all organization. An Organization Administrator only sees the locations for his/her organization. A Simple User only sees the organizations to which he/she is assigned.

###### List for Service Administrator

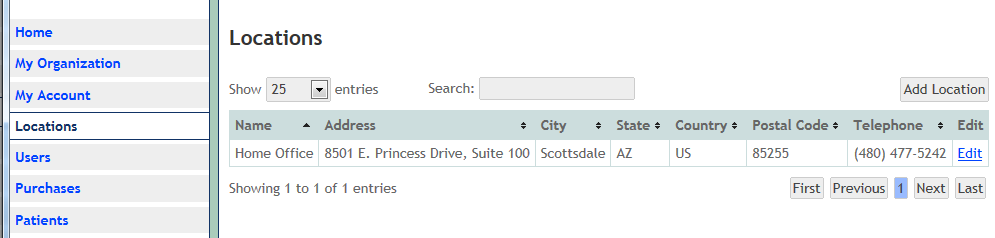
When a Service Administrator clicks the *Locations* entry in the main menu, a list of all currently configured locations is presented. This list contains the locations for all organizations and identifies each location’s organization in a column in the list. The list is initially sorted alphabetically by organization. The following image presents an example:



Since the list contains only some of the details for each location, a *View* link is provided for each location in the list. Clicking the link will allow the Service Administrator to see all the details. An *Edit* link is also provided for each location to allow modification of the location. The *Add Location* button in the upper right corner allows a Service Administrator to add a new location.

###### List for Organization Administrator

When an Organization Administrator clicks the *Locations* entry in the main menu, a list of all currently configured locations for his/her own organization is presented. This list is initially sorted alphabetically by name. The following image presents an example:

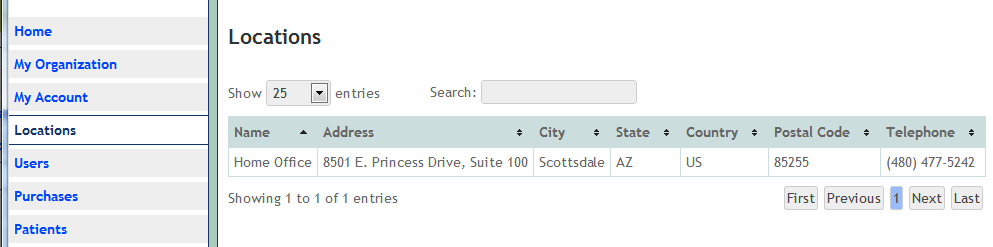


This organization has only one location so it is the only one presented. If there were more, they would be included in the list.

Since all locations are for the user’s organization, the *Organization* column is not present. Also, there is no *View* link because all the details viewable by an Organization Administrator are in the list. An *Edit* link is provided for each location to allow modification of the location. The *Add Location* button in the upper right corner allows an Organization Administrator to add a new location.

###### List for Simple User

When a Simple User clicks the Locations entry in the main menu, a list of all locations to which the user is currently assigned is presented. This list is initially sorted alphabetically by name. The following image presents an example:

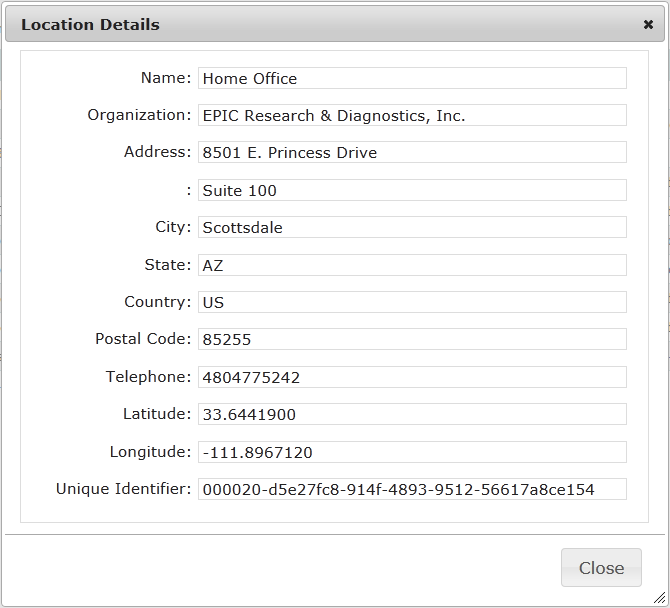


This user is assigned to only one location so it is the only one presented. If the user were assigned to multiple locations, all of those locations would be in the list.

This list is very similar to the one for an Organization Administrator. All locations are for the user’s organization, so the *Organization* column is not necessary. Notably absent are the *Edit* link for each location and the *Add Location* button. A Simple User is not allowed to add new locations or modify existing locations.

##### View Location Details

When a user clicks the *View* link for a location in the list, a pop-up dialog presents more details about the location. The details view is only provided for Service Administrators. The following image presents an example:



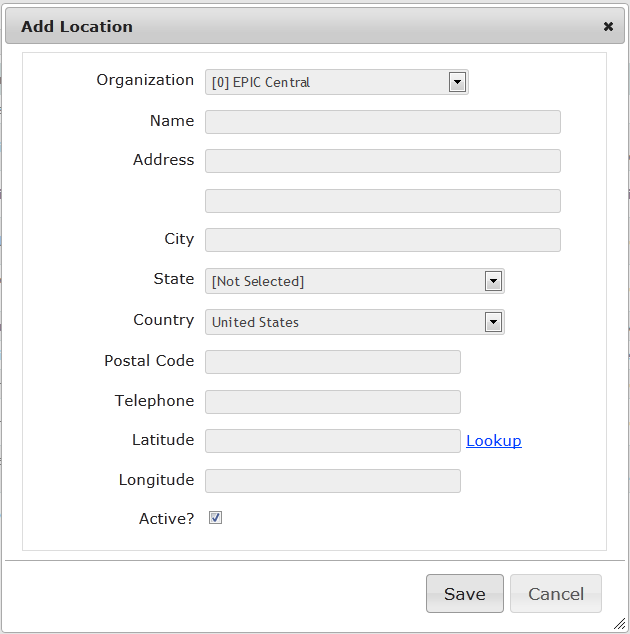
The basic information in the details view is also in the list. The details view includes the *Country*, *Postal Code*, geocoding information and the *Unique Identifier* for the location.

##### Add a New Location

The functionality provided for adding a new location is slightly different for a Service Administrator compared to an Organization Administrator. A Simple User has no ability to add a location.

###### Add for Service Administrator

To add a new location, a Service Administrator clicks the *Add Location* button presented by the table that lists the locations. When the button is clicked, the following pop-up dialog is displayed:



A Service Administrator must select the organization to which the location is being added. The name of the location is required. An address and telephone number should be provided. The user can click the *Lookup* link to find and populate the *Latitude* and *Longitude* using the address previously entered. If these values can’t be found, they will be left empty and the user can manually enter them. They are not required to save and create the location.

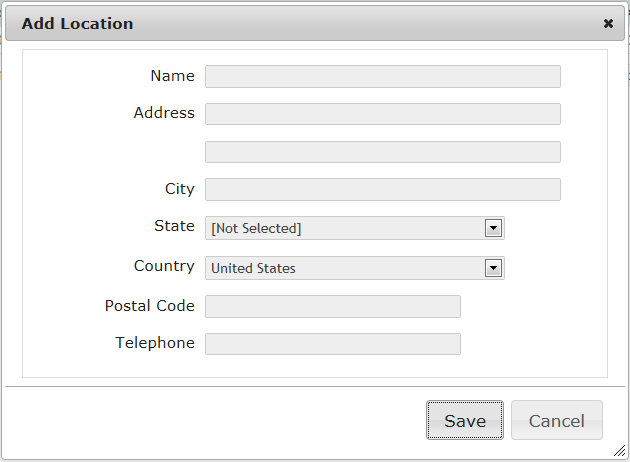
Normally, a new location is created as an active organization. Until a device is assigned to it, it will effectively be dormant since all activity starts with a device. But if the user wants to create an inactive organization, the *Active* checkbox can be unchecked.

When all input is complete, clicking the *Save* button will cause the location to be created and the dialog box removed. Clicking the *Cancel* button at any time will remove the dialog box and return to the list of locations with no modifications made.

When the *Save* button is clicked, EPIC Central will validate a name has been entered. If not, it will display an error message and allowed the user to make corrections. If the name is valid, if possible, the geocoding values will be found, a unique identifier will be generated, and the location will be saved.

###### Add for Organization Administrator

To add a new location, an Organization Administrator clicks the *Add Location* button presented by the table that lists the locations. When the button is clicked, the following pop-up dialog is displayed:



Since an Organization Administrator can only create locations for his/her own organization, there is no ability to select an organization. The name of the location is required. An address and telephone number should be provided.

When all input is complete, clicking the *Save* button will cause the location to be created and the dialog box removed. Clicking the *Cancel* button at any time will remove the dialog box and return to the list of locations with no modifications made.

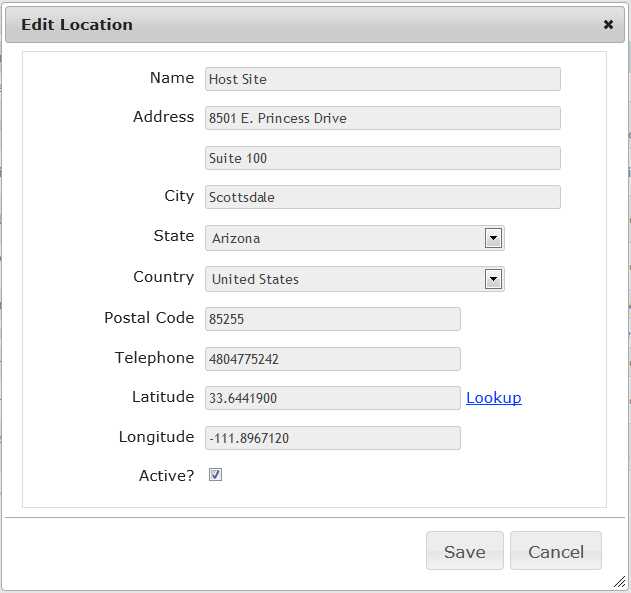
When the *Save* button is clicked, EPIC Central will validate a name has been entered. If not, it will display an error message and allowed the user to make corrections. If the name is valid, if possible, the geocoding values will be found, a unique identifier will be generated, and the location will be saved.

##### Edit an Existing Location

The functionality provided for modifying an existing location is also slightly different for a Service Administrator compared to an Organization Administrator. A Simple User has no ability to modify a location.

###### Edit for Service Administrator

To edit a location, a Service Administrator clicks the *Edit* link in the rightmost column of the row that presents the location to be modified. When the link is clicked, the following pop-up dialog is displayed:



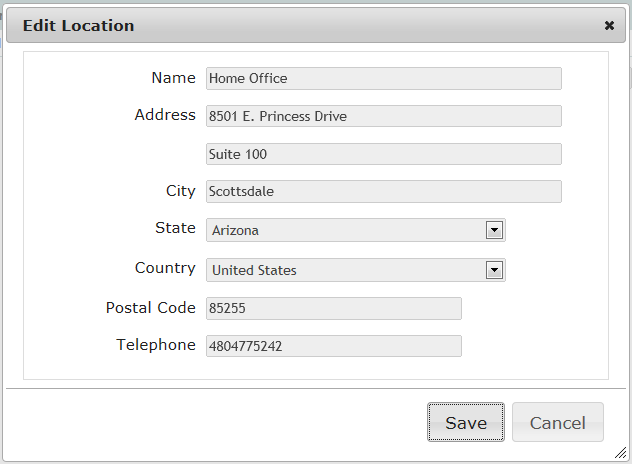
A Service Administrator can change the name, address or telephone number for the location. The *Latitude* and *Longitude* normally cannot be changed. However, if the *Lookup* link is clicked, the values will be found again and the fields filled. The fields will then be editable. This functionality is provided so the user can find the new values when changing the address.

A Service Administrator can also make a location inactive by unchecking the *Active* checkbox, or make it active again by checking it.

When the *Save* button is clicked, EPIC Central will validate a name has been entered. If not, it will display an error message and allow the user to make corrections. Otherwise, if the address has changed, it will attempt to find the new geocoding information to update the values. If the address hasn’t changed, no geocode query will be done. The location update will be saved.

###### Edit for Organization Administrator

To edit a location, an Organization Administrator clicks the *Edit* link in the rightmost column of the row that presents the location to be modified. When the link is clicked, the following pop-up dialog is displayed:



An Organization Administrator cannot manually look up or edit the geocoding information, or change the *Active* flag, so those fields are not provided. Otherwise, an Organization Administrator can change the name, address and telephone number just like a Service Administrator. And the save functionality is the same.

#### Data

The Location table in the database contains a record for each location defined in EPIC Central. When a new location is created, another record is added to the table. When an existing location is modified, the existing record is updated.

No records are ever deleted from the Location table. Making a location inactive effectively removes it from the system. Devices for an inactive location are not allowed to authenticate and if a user is assigned only to inactive locations, the user cannot log in. *[Req: 4.7.3.1]*

### User

An MVC component that provides the ability for a Service Administrator to view, create and modify user accounts for any organization, an Organization Administrator to view, create and modify user accounts for his/her own organization, and any other user to view the users at all locations to which he/she is assigned.

User is also a data object and a table in the database. Users are attached to organizations and “assigned” to locations within that same organization. Non-administrator users are only allowed access to locations they are assigned to. Support issues, credit cards and purchases are associated with users.

A user has a single role which defines the access rights/permissions for accessing functions and content. A user’s role is set by an administrator. Only a Service Administrator can assign the Service Administrator role to another user.

#### Purpose

The User MVC component exists to provide the ability for administrators to administer users and for other users to view existing users assigned to the same locations. Any person who requires access to EPIC Central must have a user account. *[Reqs: 4.1.4.1, 4.1.4.2.1]*

A Service Administrator is able to view all the users currently defined for all organizations and make modifications to them. A Service Administrator is also able to add a new user for any organization and assign any role to any user. *[Reqs: 4.1.4.2.2, 4.1.5.4.1, 4.1.5.4.2]*

An Organization Administrator is able to view all the users currently defined for his/her own organization and make modifications to them. An Organization Administrator is also able to add a new user account for that organization and assign either the Organization Administrator or Simple User roles. *[Reqs: 4.1.4.2.3, 4.1.5.4.1, 4.1.5.4.3]*

When creating a new user account, an administrator must provide the user’s first name, last name and email address. The administrator must also set the user’s role and organization and assign the user to locations in the organization, if the user is a non-administrator user; administrators have access to all locations. EPIC Central will validate the format of the email address and, if all entries are valid, it will send a registration email to that address. The user must use that email to complete a registration process to activate the account. The email will contain a key to identify the account waiting for registration to be completed. Until the registration process is completed, the user account will not be activated and is unusable. *[Reqs: 4.1.4.2.4, 4.1.4.2.5, 4.1.4.2.6, 4.1.4.2.7, 4.1.4.2.8, 4.1.4.2.9, 4.1.5.2, 4.1.5.4.5]*

When editing a user account, an administrator can change the user’s user name, first name, last name and email address, or assign the user a different role. Any role change takes effect on the next action a user performs. A Service Administrator can also make a user inactive which will prohibit the user from logging in. *[Reqs: 4.1.5.4.6, 4.1.5.4.7]*

#### Function

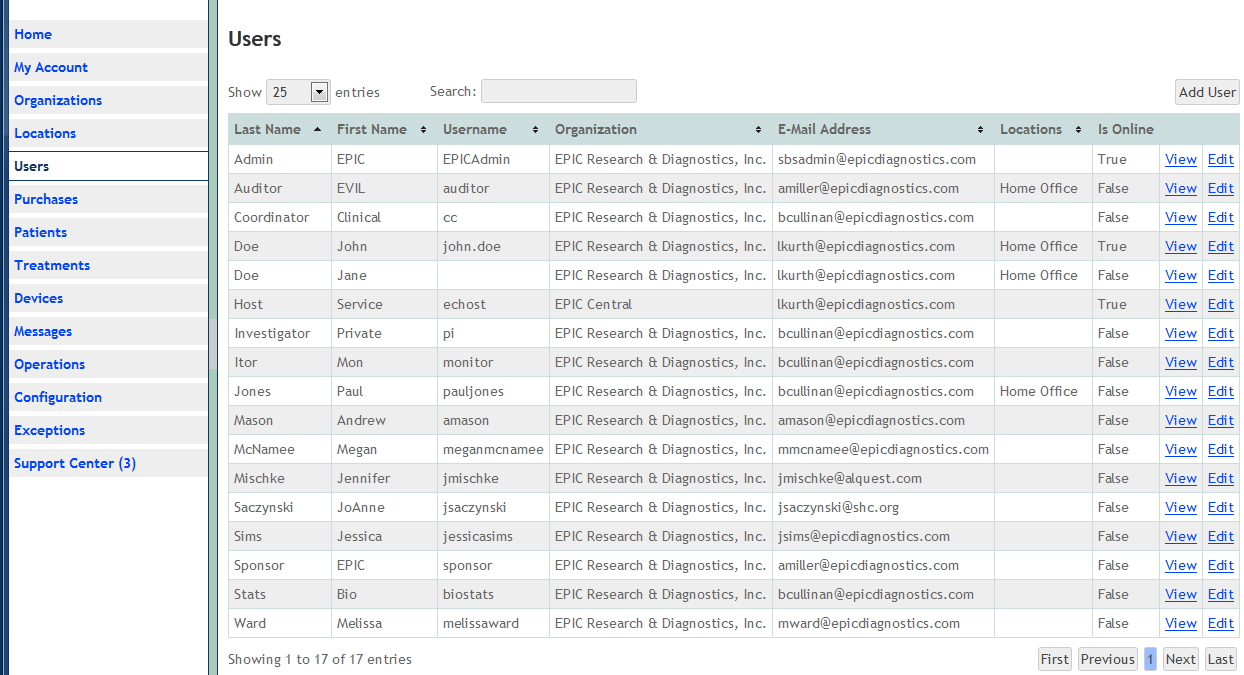
This component provides all the necessary functionality to view and administer users. These functions include the following:

##### List of Existing Users

The list of other users a user sees is dependent on the user’s role. A Service Administrator can see all the users for all organizations. An Organization Administrator only sees the users for his/her own organization. A Simple User only sees other Simple Users assigned to the same locations as he/she is assigned.

###### List for System Administrator

When a Service Administrator clicks the *Users* entry in the main menu, a list of all currently configured user accounts is presented. This list contains the user accounts for all organizations and identifies each user’s organization in a column in the list. The list is initially sorted alphabetically by *Last Name*. The following image presents an example:

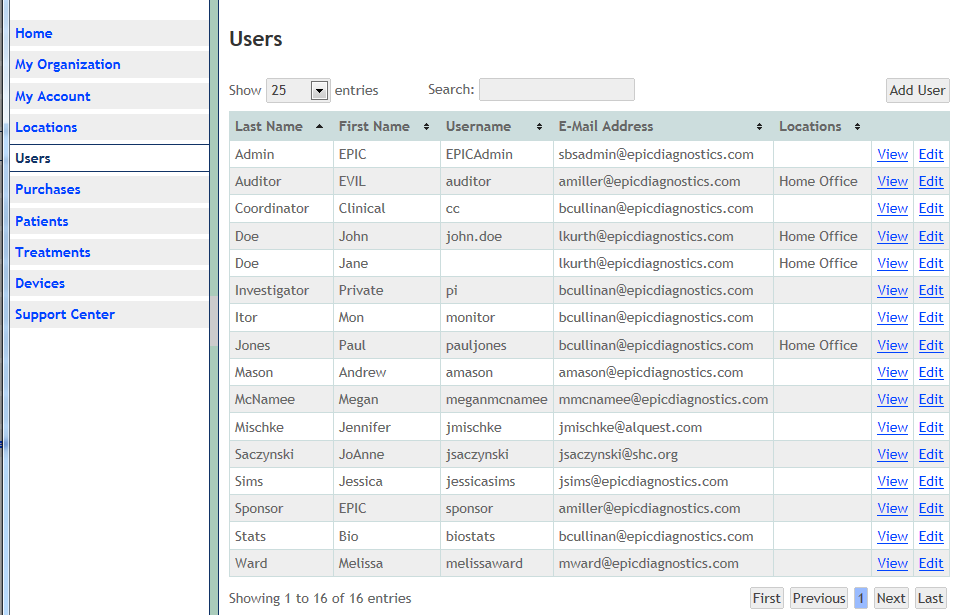


Since the list contains only some of the details for each user, a *View* link is provided for each user in the list. Clicking the link will allow the Service Administrator to see all the details. An *Edit* link is also provided for each user to allow modification of the user. The *Add User* button in the upper right corner allows the Service Administrator to add a new user account.

An empty entry in the *Username* column is indicative of a new user who has not completed registration; a username is chosen during the registration process. An empty entry in the *Locations* column is indicative of an administrator user; administrators are not assigned to locations, instead they can see all locations for their organization.

###### List for Organization Administrator

When an Organization Administrator clicks the *Users* entry in the main menu, a list of all currently configured user accounts for his/her organization is presented. This list is initially sorted alphabetically by *Last Name*. The following image presents an example:

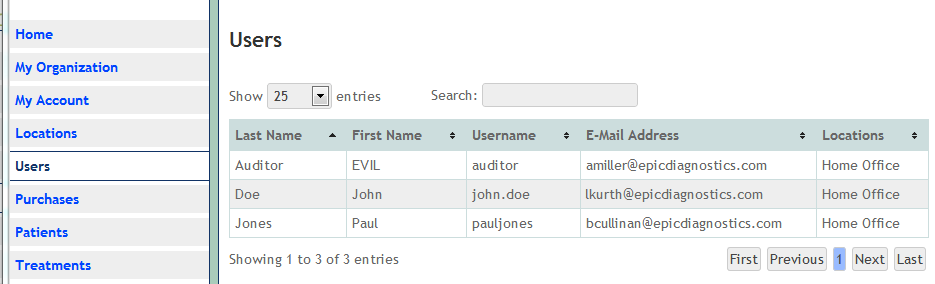


Since all users are associated with the Organization Administrator’s organization, the *Organization* column is not presented. Also missing, compared to the Service Administrator’s list, is the *Is Online* column; that column is exclusive to Service Administrators.

Like the list for Service Administrators, the *Username* or *Locations* entries for a user may be empty; these mean the same as in the other list. Also, there are *View* and *Edit* links to view more details or to make modifications, respectively.

###### List for Simple User

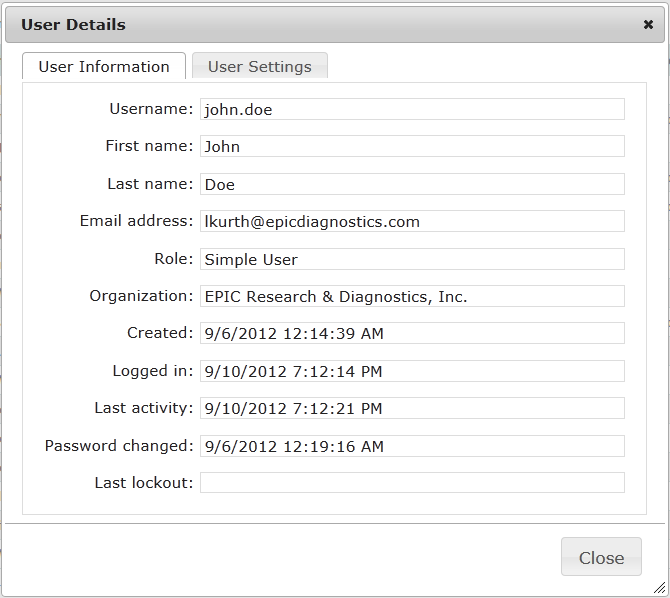
When a Simple User clicks the *Users* entry in the main menu, a list of all the Simple Users assigned to the same locations as the logged-in user is displayed. The list is initially sorted by *Last Name*. The following image presents an example:



The columns are the same as for the Organization Administrator’s list except there are no *View* or *Edit* links. A Simple User can only view the information in the columns of the list and cannot make changes to another user.

##### View User Details

When a user clicks the *View* link for a user account in the list, a pop-up dialog presents more details about the user account. The details view is the same for Service Administrators and Organization Administrators. The following image presents an example:

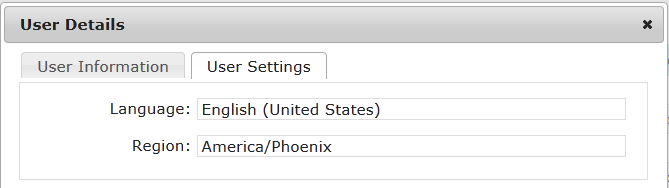


In addition to the columns in the list, the *User Information* tab of the details view shows the user’s role and several date/time values related to activities on the user account:

* *Created* – The date and time when the user account was created by an administrator.
* *Logged in* – The date and time when the user last logged in.
* *Last activity* – The date and time of the last activity the user performed while logged in.
* *Password changed* – The date and time when the user’s password was last changed.
* *Last lockout* – The date and time when the user was last locked out of EPIC Central.

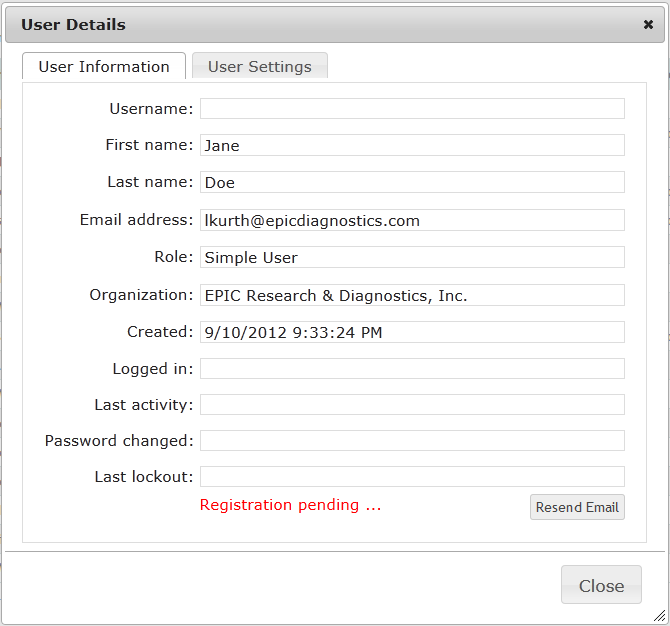
If any of the date/time values is not set, it will simply be shown with no value.

The *User Settings* tab shows the user’s settings for *Language* and *Region*. These determine the presentation language for the user interface and the region/locale/time zone for date display. The following image presents an example:



###### Unregistered User Details

If the user has not yet completed the registration process, the details view is slightly different. The following image presents an example:



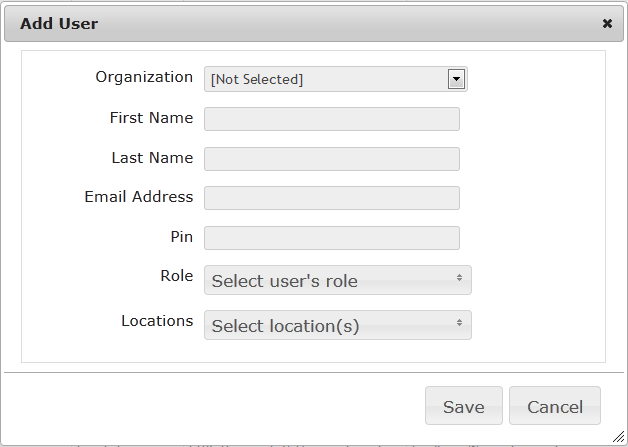
The *Username* and several of the date/time fields are empty because they have not been set. Also, at the bottom is a message to indicate that the user’s registration is pending. Until the user completes the registration, the administrator can resend the registration email by clicking the *Resend Email* button.

##### Add a New User

The functionality for adding a new user account is slightly different for a Server Administrator compared to an Organization Administrator. A Simple User has no ability to add a user account.

###### Add for System Administrator

To add a new user account, a Service Administrator clicks the *Add User* button presented by the table that lists the existing users. When the button is clicked, the following pop-up dialog is displayed:



The Service Administrator must select the *Organization* to which the user account will be added. When the organization is selected, the *Role* and *Locations* select lists will be populated depending on the organization. If the organization selected is the service host (EPIC Central) the role list will contain all three roles: Service Administrator, Organization Administrator and Simple User. If any other organization is selected, the Service Administrator role will not be included.

The user’s *First Name*, *Last Name* and *Email Address* are all required fields. The email address must have a valid format. This is the address where the registration email will be sent.

The *PIN* is any string of characters known to both the administrator creating the account and the person who will register the account. This is effectively a shared secret and helps prevent someone from hijacking the account during the registration process. The *PIN* is not included in the registration email. It must be shared by some mechanism outside the boundaries of EPIC Central.

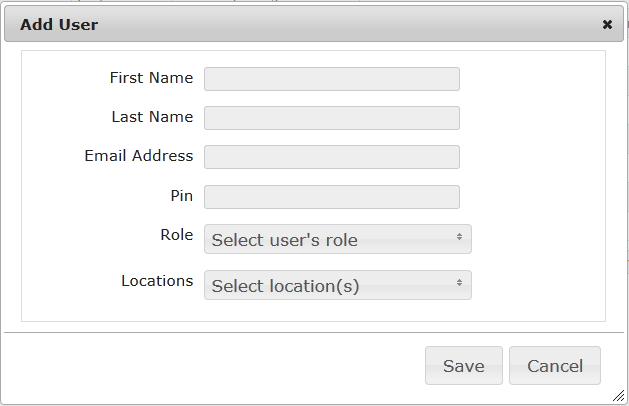
The Service Administrator must select the user’s role. If the role is set to Simple User, then the administrator can select the locations the user will be assigned to. If any other role is selected, the *Locations* select list is disabled; administrator users have access to all locations so they are not assigned to specific locations.

When all input is complete, clicking the *Save* button will cause the user account to be created and the dialog removed. Clicking the *Cancel* button at any time will remove the dialog box and return to the list of existing user accounts with no modifications made.

When the *Save* button is clicked, EPIC Central will validate that *First Name*, *Last Name*, *Email Address* and *PIN* are present. It will validate the format of the email address. It will validate that an *Organization* is selected, a *Role* is selected and *Locations* selected if the role is Simple User. If all validations pass, it will create a new user account, assign the user to the appropriate locations, if necessary, and send a registration email to the email address specified. If one or more validations fail, an error message will be presented and the user will be allowed to make corrections.

###### Add for Organization Administrator

To add a new user account, an Organization Administrator clicks the *Add User* button presented by the table that lists the existing users. When the button is clicked, the following pop-up dialog is displayed:



This dialog is the same as the one for a Service Administrator, except this one does not contain the selection for *Organization*. Since an Organization Administrator can only add users to his/her own organization, the organization cannot be changed. Also, the *Role* selection does not contain the Service Administrator role; only another Service Administrator can assign that role.

There is one other important difference. Since the organization is known, if the organization has only one location, the *Locations* selection will not be shown. If the new user’s role is Simple User, the user will just be assigned to that one location. If there is more than one location, the *Locations* selection will be shown in the form.

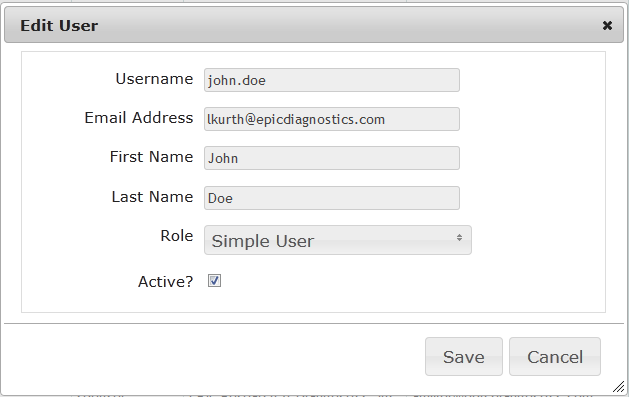
The *Save* and *Cancel* functions are the same as for a Service Administrator. Validation is the same except there is no organization to be validated.

##### Edit an Existing User

The functionality provided for modifying an existing user account is also slightly different for a Service Administrator compared to an Organization Administrator. A Simple User has no ability to modify another user’s account.

###### Edit for System Administrator

To edit an existing user account, a Service Administrator clicks the *Edit* link for a user in the user account list. When the link is clicked, a pop-up dialog containing the user’s configuration will be presented. The following image shows an example:

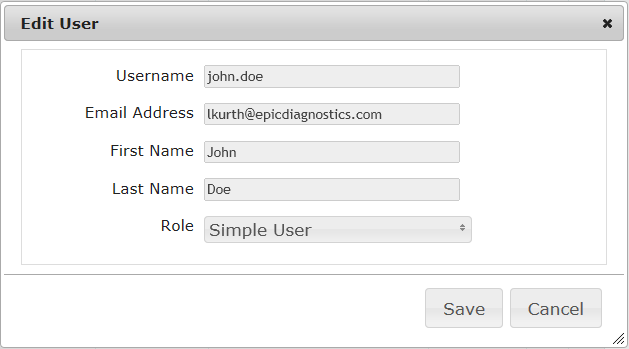


The Service Administrator can edit any of the fields, change the user’s role, or make the user inactive or active by changing the state of the *Active* checkbox. Notably absent is the ability to change the organization a user belongs to. That is not allowed.

When the *Save* button is clicked, EPIC Central will validate the username is present and unique among all users, the email address is present and has a valid format, and the first and last names are present. If all fields properly validate, the changes are saved to the user account record. If one or more validations fail, an error message is presented and the user is allowed to make corrections. At any time, if the *Cancel* button is clicked, the dialog box is removed and no modifications made.

###### Edit for Organization Administrator

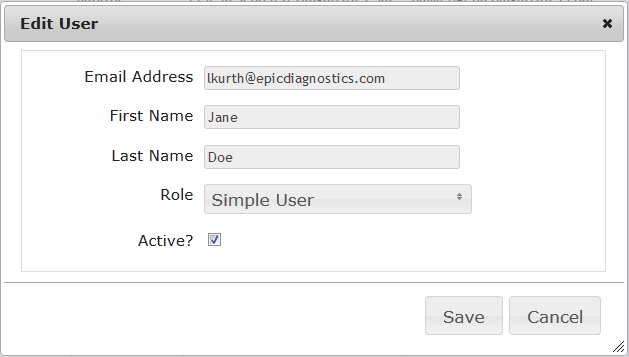
To edit an existing user account, an Organization Administrator clicks the *Edit* link for a user in the user account list. When the link is clicked, a pop-up dialog containing the user’s configuration will be presented. The following image shows an example:



The only difference from the Service Administrator’s functionality is an Organization Administrator cannot change the user’s active state. All other functionality is identical.

###### Edit a Pending Registration

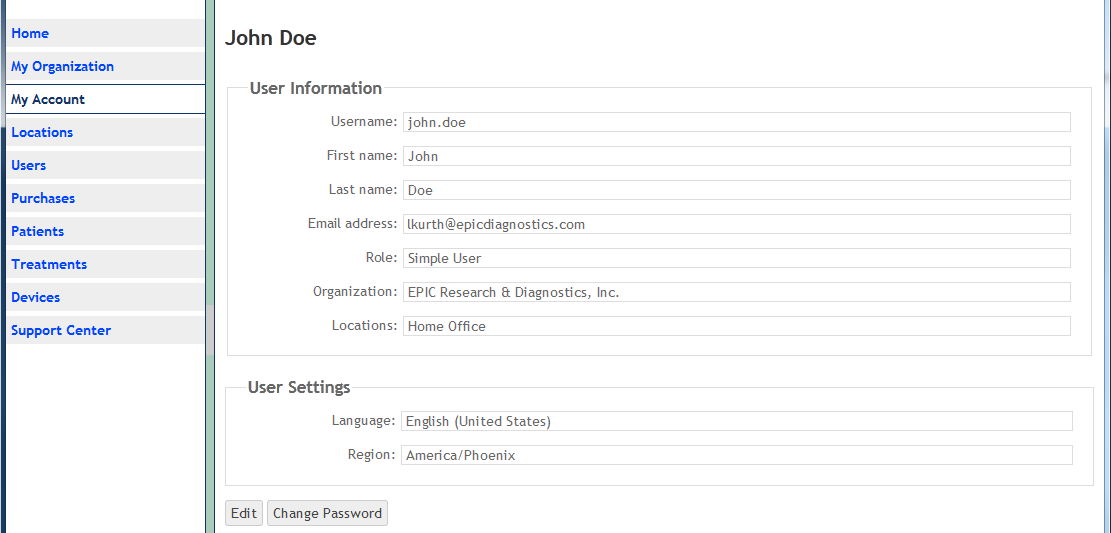
If a user account has been created, but the user has not completed the registration process, there is no user name to edit. The following image shows an example:



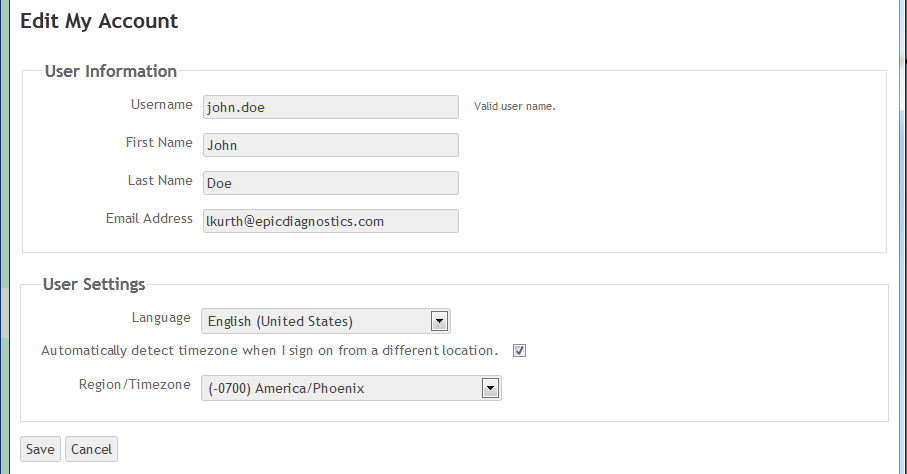
This is a Service Administrator’s view. An Organization Administrator’s view will not include the *Active* checkbox. Validations and save functionality are the same as for editing a fully registered user.

##### Edit One’s Own Settings

Any user is able to edit some of his/her settings. This functionality is the same for all users and is accessed through the *My Account* entry in the main menu. The following image shows the *My Account* information page:



The user can click the *Edit* button at the bottom to make changes to some settings. When the *Edit* button is clicked, an edit form is presented with the settings initialized to the user’s current settings. The following image shows an example:



In the upper area of the form, the user can change his/her user name, first and last names and email address. When the user name is changed, after each character change the user name is sent to EPIC Central for validation. User names must be at least 8 characters long and cannot contain the space character. They must also be unique. As the user name is changed, a hint message is placed to the right of the textbox to indicate whether or not it is valid, and if it’s invalid what the user needs to do to make it valid. This control is described in more detail below in section *6.2.4.2.3, User Registration*.

In the lower area of the form the user can change settings that affect the user experience on the site. When a user registers and logs in the first time, the language is detected from the accept headers provided by the browser and the *Language* setting is automatically set. For example, if the user is in Germany and uses a German-language browser, the *Language* setting would be initialized to German. If the language detected is not found as a language supported by EPIC Central, it will default to US-English. At any time, the user can come to this form to change the language to any language supported by EPIC Central. This setting remains in effect until the user changes it again. *[Reqs: 4.8.3.1, 4.8.3.3, 4.8.3.4]*

EPIC Central has the ability to automatically detect the user’s time zone based on the time zone setting of the operating system where the browser is running. To allow a user to roam to different time zones and use the current time zone wherever the user happens to be, the checkbox should be checked. In this case the *Region/Timezone* selection will show the current time zone. If a user prefers to always use the same time zone, even when accessing EPIC Central from another time zone, then the checkbox should be unchecked. The user can then set the time zone using the *Region/Timezone* selection and it will be set until the user changes these settings again.

#### Data

The User table in the database contains a record for each user account defined in EPIC Central. When a new user account is created, another record is added to the table. When an existing user account is modified, the existing record is updated. Each User is related to only one entry in the Organization table.

No records are ever deleted from the User table. Making a user account inactive effectively removes the user from the system. An inactive user cannot authenticate to perform any function. *[Req: 4.7.3.1]*

The UserRole table is used to join a user to a role. The roles are predefined in the Role table. A user has only one role, so there will always be one entry in the UserRole table for each user. When a user’s role is changed, the currently entry is deleted and a new entry created to join the user to the new role.

The UserAssignedLocation table is used to join (assign) users to locations. Entries are created whenever a user is assigned to a location and deleted when the user is unassigned.

The UserSalt table contains a salt value for each User to strengthen the encryption of passwords. The salt is a random string of characters. It is appended to the user’s password prior to encryption and the encrypted value stored as the password in the User table. A one-way SHA-512 encryption algorithm is used. This means that the actual password a user enters is never stored by EPIC Central. Thus, it cannot be recovered. If a user forgets his/her password, he/she must go through a reset process and create a new password.

The UserSetting table contains settings for each user. The settings can be anything the application needs; it is simply a name/value pairing. A couple of the settings are the language for the UI and the region/locale mentioned about in the user details view.

### Account

An MVC component that provides user account related functions, including log on, log off, registration and password management. These functions are available for use by all users in any role. There are no differences based on role.

#### Purpose

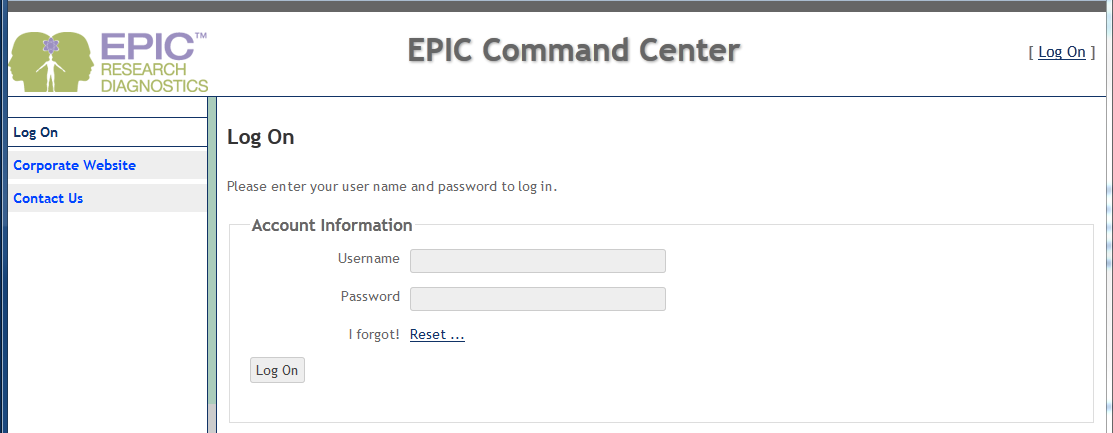
The purpose of this component is to hold various functions the user needs for managing his/her own account. This includes the ability to log on and log off, change a password, and reset a password if it or the user name is forgotten.

#### Function

This component allows users to log on to EPIC Central, log off, and to manage their passwords. These specific functions are explained in the following sections.

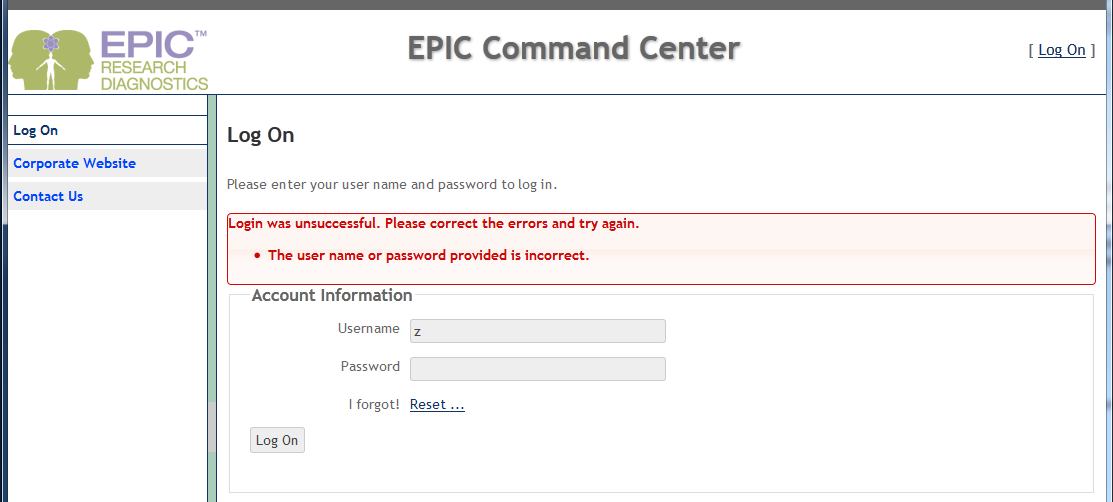
##### Log On

When an unauthenticated user attempts to access any page or function of the EPIC Central site, the user will be redirected and land on the *Log On* page. The following image shows the basic page:



The user must enter his/her user name and password and click the *Log On* button to log on to EPIC Central. The *Corporate Website* and *Contact Us* entries in the main menu on the left are links to the EPIC corporate website. The *Reset* link will take the user to a page where he/she can reset the password on the account. That function is described below. *[Req: 4.2.1.1]*

If the user incorrectly enters the user name or password, an error message will be presented and the user can make corrections and try again. The following image shows the page with the error message present:



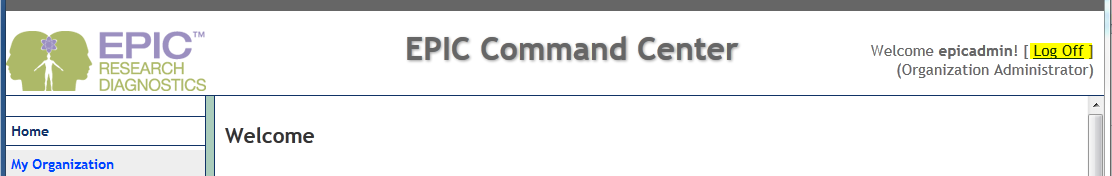
If either of the fields is empty when the *Log On* button is clicked, the error message will indicate that the field is required.

On successful log on, the user will normally land on the *Home* page of the EPIC Central site, or the page they tried to access immediately prior to authentication.

There is no ability for the user to save log on credentials in the browser’s cache. The user name and password must be reentered every time to log on. *[Req: 4.2.1.2]*

##### Log Off

An authenticated user can log off any time by simply clicking the *Log Off* link in the upper right corner of every page. The following image highlights the *Log Off* link:



When the user clicks the *Log Off* link, EPIC Central will close the user’s session and clear the authentication cookie. The user will be redirected to the *Log On* page (see above) where the user can log in again. *[Reqs: 4.2.1.3, 4.2.1.4]*

If an authenticated user performs no functions within a 60-minute time period, the user will be automatically logged off and the session cleared. The inactivity timeout value will initially be 60-minutes, but it is configurable. *[Req: 4.2.1.5]*

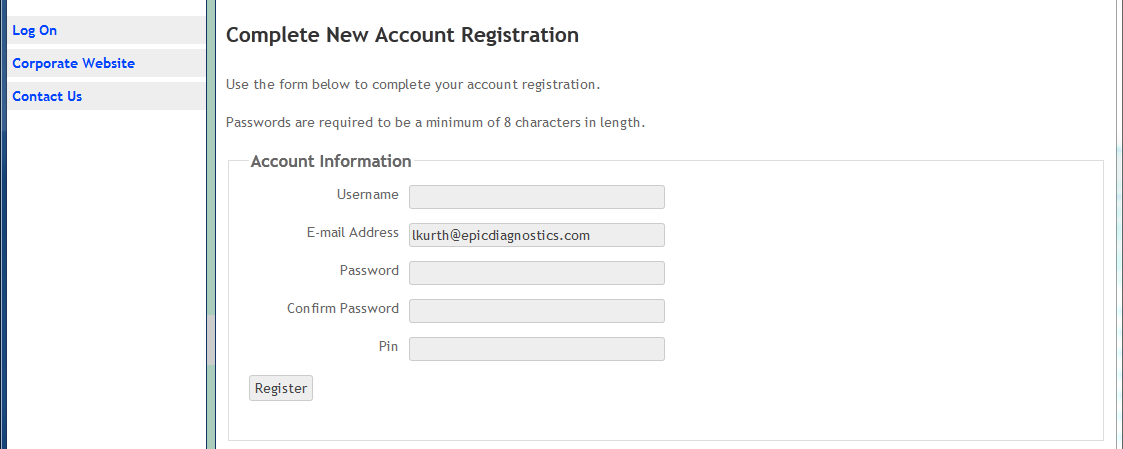
##### User Registration

When an administrator creates a user account, the account is placed into a restricted state until the user comes to EPIC Central and registers the account. Registration requires the user to select a user name and password. *[Reqs: 4.1.4.3.1, 4.1.4.3.3, 4.7.1.1, 4.7.1.2, 4.7.1.3, 4.7.1.4, 4.7.1.5]*

When the account is created, an email is sent to the user at the email address entered by the administrator. The following image presents an example of a registration email message:



The user begins the registration process by clicking the *Complete Registration* link in the body of the email message. The link contains two parameters: the email address and a unique registration key that identifies the user account created by the administrator. *[Req: 4.1.4.3.2]* EPIC Central uses the parameters to locate the user account and confirm it is a new account that needs to be registered. It then presents the *Complete New Account Registration* page. The following image presents an example:

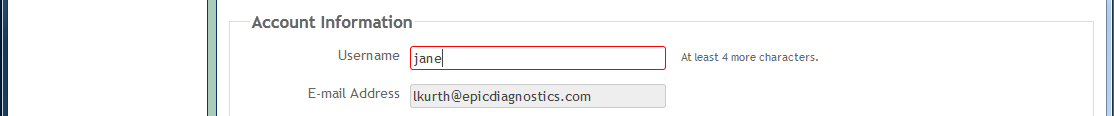


If any error occurs, an error message is presented instead. For example, if the user registers the account and then clicks the link in the email again, there will be an error because the account cannot be registered twice.

To complete the registration, the user must:

* Enter a valid user name. *[Req: 4.1.4.3.3]*
* Confirm the email address for the account, or enter a different one.
* Enter a valid password. *[Req: 4.1.4.3.4]*
* Enter the password again to confirm it. *[Req: 4.1.4.3.5]*
* Enter the same PIN the administrator entered when the account was created.

User names can contain any character except a space. They must be at least 8 characters long. They must also be unique among all users. As the user enters a user name, the system will validate it after each character is entered. *[Req: 4.1.4.3.6]* It will display a hint to the right of the textbox to indicate what the user needs to do to make the user name valid. The following image shows an example:

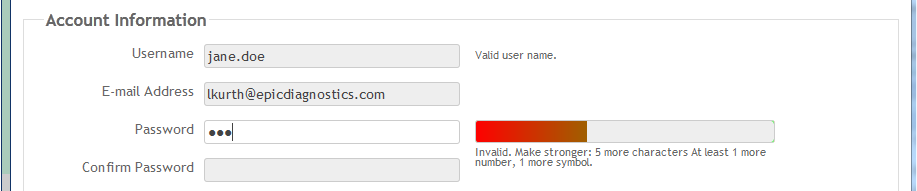


The textbox is highlighted in red to indicate the user name is not valid. The hint tells the user what to do. When a valid user name is entered, the red highlight will disappear and the hint will become “Valid user name.” Once the user name is valid, if the user continues to enter characters and the user name is not unique, the hint will become “User name is already in use.” The user must change the user name to make it valid.

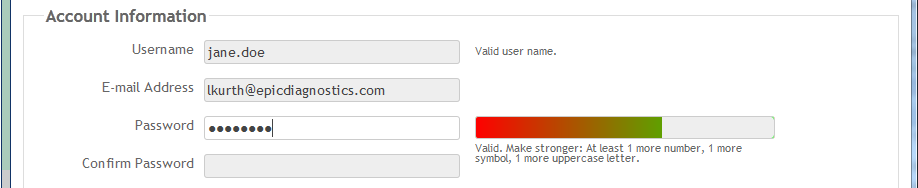
The email address in the form is initially set to the email address entered when the administrator created the account. The user can change this address if he/she wants to use a different email address for the account. The email address must have a valid format.

The user must enter a password that meets the requirements for password structure. Currently, passwords are required to be 8 to 24 characters long and must include one uppercase letter, one lowercase letter, one numeric digit and one special symbol. The space character is not allowed. (Note: This is slightly more strict that the requirement because it requires a character from each of the four character types.) *[Req: 4.1.4.3.7]*

As the user enters a password, a “strength meter” is displayed to the right of the password textbox. The meter begins with a red color and turns to green on the right when the password becomes a strong password. A hint message below the meter tells the user the password is invalid and what to do to make it valid. The following image shows an example:



Once the password entered is valid, the hint message will change to indicate the password is valid and provides the user with additional advice on how to make it stronger. The following image shows an example:



The user can continue to enter more characters to make the password stronger. Once it reaches what EPIC Central considers a strong password, the meter will fill with green all the way to the right end and the hint message will be “Valid. Strong password!” If the user enters a space character the hint message will indicate no spaces are allowed. If the password gets too long, the hint message will be “Invalid. Password too long!”

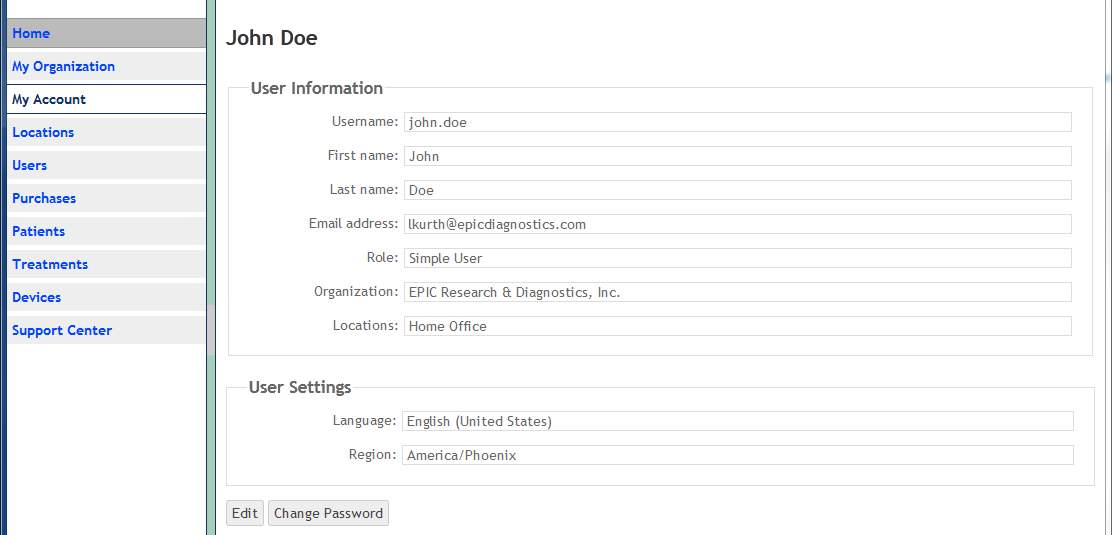
To confirm the user remembers the password entered, the user must reenter the password in the *Confirm Password* textbox. No meter is shown. Once the confirm textbox receives focus, it will be highlighted in red until the two passwords match. The user will receive an error if the two passwords don’t match when the *Register* button is clicked. *[Req: 4.1.4.3.8]*

Lastly, the user must enter the PIN the administrator entered when the account was created. This PIN is something the user must know, or receive from the administrator via some communication means other than EPIC Central. The PIN is not included in the email.

Once all entries are made, the user must click the *Register* button to complete user registration. EPIC Central will revalidate all the fields entered. If they are valid, it will set the user name, password and email address for the account and log the user in. The user will be redirected to the *Home* page. If any field is invalid, an error message will be displayed in the page and the user can make corrections and resubmit. *[Reqs: 4.1.4.3.9, 4.1.4.3.10]*

##### Change Password

A user can change his/her password at any time. *[Req: 4.2.6.1.1]* To change a password, a user must select the *My Account* entry in the main menu. This will display the user’s account information and include a *Change Password* button at the bottom. The following image shows an example:



When the user clicks the *Change Password* button, a form will be presented where the user can entered the current password and a new password. The following image shows an example:



The *Old Password* is the user’s current password. This will be validated when the form is submitted. The user must enter a new password and also confirm it. The password strength meter described in the previous section is also used in this form. It works identically. The *Confirm Password* textbox works the same as described in the previous section, too.

Once the user has entered the old/current password, the new password and confirmed it, the user clicks the *Save* button to save the new password. EPIC Central will validate all the fields and if they are valid set the password for the account to the new password. The next time the user needs to log in, the new password will be required. If any field is invalid, an error message is displayed in the page and the user can make corrections. *[Reqs: 4.2.6.1.2, 4.2.6.1.3, 4.2.6.1.4]*

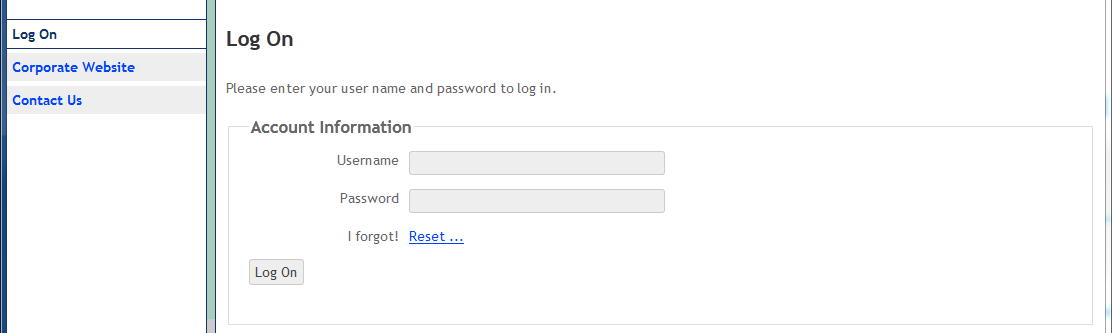
At any time, the user can click the *Cancel* button to return to the *My Account* information page with no modifications made.

##### Password Reset

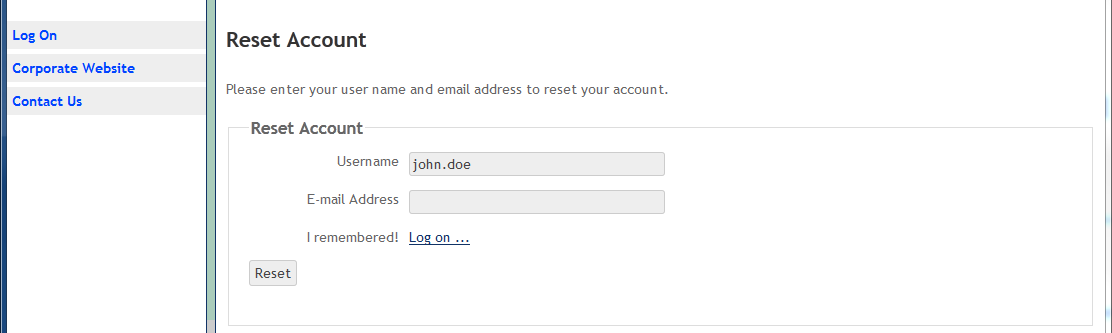
If a user forgets his/her user name and/or password, the user can request that the password for the account be reset. Since EPIC Central does not store passwords, only a secure hash of each password, it cannot provide the user’s password to the user. The user must go through a reset process to reset the password to a known value. *[Req: 4.2.6.2.1]*

###### Starting the Reset Process

To begin the reset process, the user clicks the *Reset* link in the *Log On* page. The following image shows the *Log On* page:



When the user clicks the *Reset* link, the user is taken to the *Reset Account* page where the user can provide the user name, if known, and email address for the account. If the user provides the user name in the *Log On* form before clicking *Reset*, the user name will be initialized in the *Reset Account* form. Otherwise, it will be empty. The following image shows an example with the user name in the form:



The email address is required and must match the email address for an existing user. If no match is found, no reset will be done. EPIC Central will only send a reset email to the email address on a user account. *[Req: 4.2.6.2.2]*

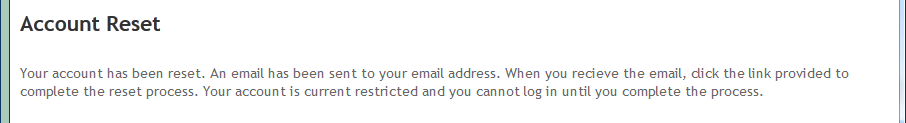
If the user doesn’t know his/her user name, it can be left empty. But the user must know the email address for the account. If both the user name and email address are provided, they must match the one and only account identified by that user name. *[Req: 4.2.6.2.3]*

After the user has provided the email address and clicked the *Reset* button, and EPIC Central will validate that a user account exists for the email address provided, and if the user name is provided, it also matches the account. If the entries are valid, it will restrict the account to prevent log on and send a reset email to that email address. The user must follow the instructions in the email to create a new password. *[Reqs: 4.2.6.2.4, 4.2.6.2.5]*

If either field has invalid data, an error message will be displayed in the page telling the user what the issue is. The user can make corrections and resubmit to attempt the reset again.

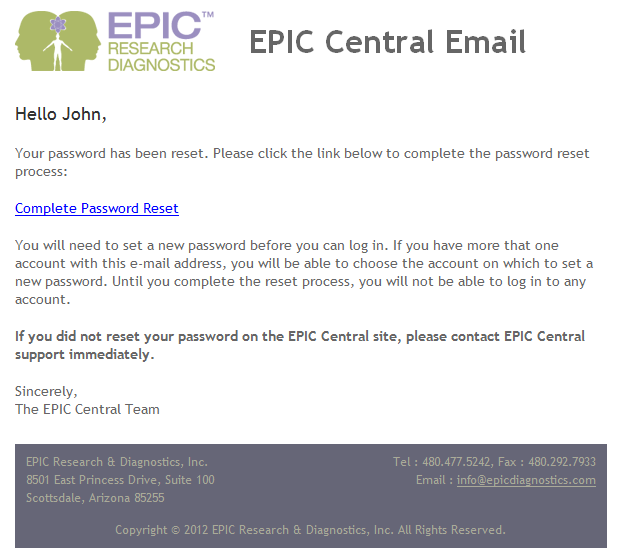
The *Log on* link in the page will take the user back to the *Log On* page and allow the user to log on.

After the user has clicked the *Reset* button and the email is sent, EPIC Central presents the following message in the page:

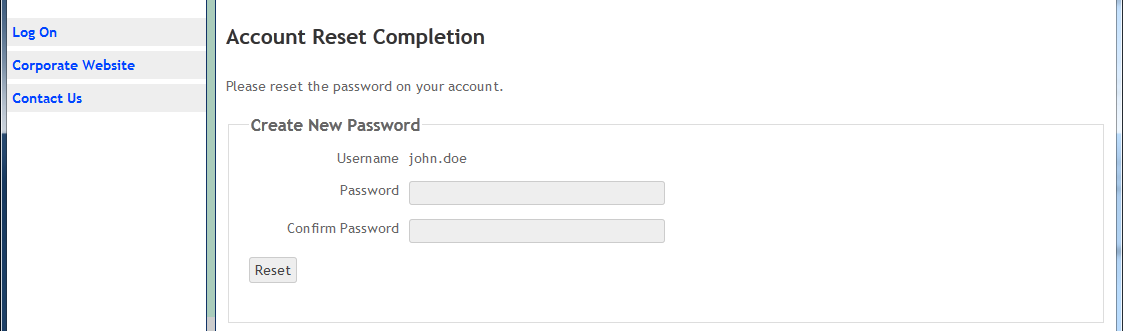


###### Completing the Reset Process

After a user resets the account using the EPIC Central form, the user will receive an email at the address for the account. The following image shows an example reset email:

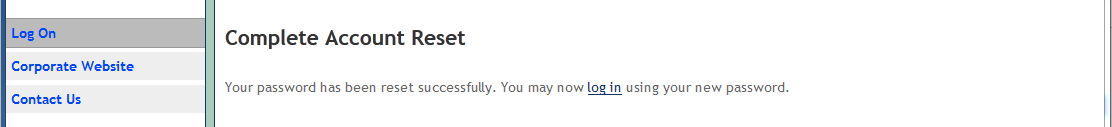


To complete the reset process, the user must click the *Complete Password Reset* link in the email to return to EPIC Central. *[Req: 4.2.6.2.7]* The link contains two parameters: a “reset key” that identifies a record of the account reset, and the email address for the account. When the link is clicked, EPIC Central will validate the key and email address before presenting the user with a page to complete the reset process. The following image shows an example of the *Account Reset Completion* page:



The user name is displayed. If the user had forgotten his/her user name, this will show the user what the user name is. There are two textboxes where the user can now enter a new password for the account and reenter it to confirm that the user knows what it is. The password strength meter described in the section *6.2.4.2.3, User Registration*, is also used in this form. It works identically. The *Confirm Password* textbox works the same as described in that section, too. *[Req: 4.2.6.2.8]*

After the user has successfully entered a new password and confirmed it, the user will click the *Reset* button to submit the new password. EPIC Central will revalidate the password entries and if they are valid, it will set the password for the account and present the following message to the user:



The user can click the *log in* link in the page or the *Log On* entry in the main menu to go to the *Log On* page to log in using the new password. *[Req: 4.2.6.2.9]*

###### Starting the Reset Process, Multiple Accounts

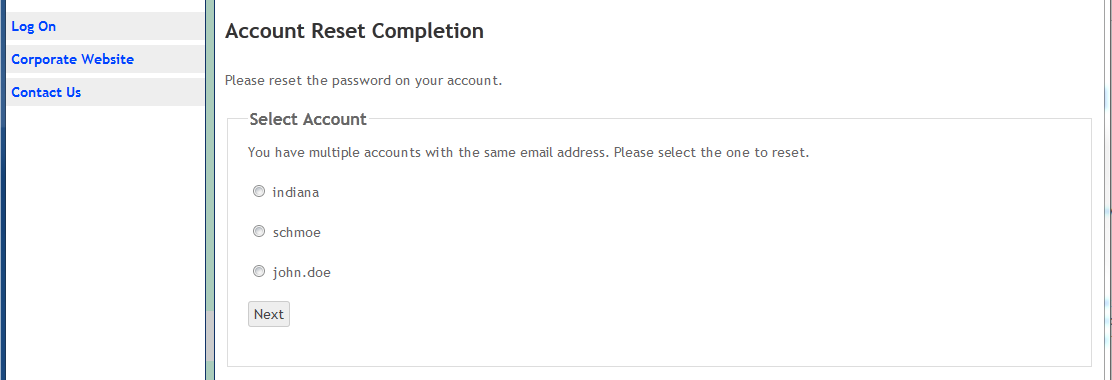
EPIC Central allows multiple accounts to share the same email address. If a user has multiple accounts with the same email address and wants to reset the password on one of them, the user must provide the user name on the *Reset Account* page as shown above. In this case, the reset process works the same as defined above.

If the user does not know the user name of the account, the user can still perform the reset process. But in this case, **all** accounts with the specified email address will be restricted until the reset process is completed. Resetting the password for any one of the accounts will remove the restrictions from all of them. Otherwise, this process is the same as outlined above. *[Req: 4.2.6.2.6]*

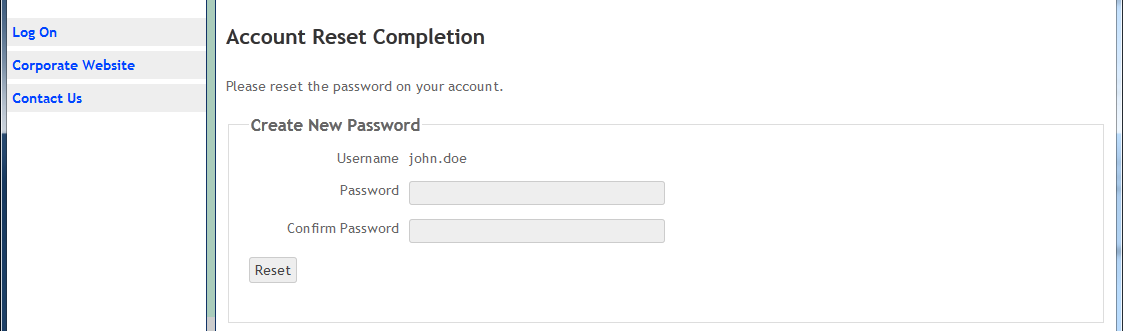
The email that is sent is identical except since the user isn’t known, it says “Hello EPIC Central User” instead of “Hello <user name>.”

###### Completing the Reset Process, Multiple Accounts

Just like the case where there is a single account reset, the user must click the *Complete Password Reset* link in the email to return to EPIC Central and set a new password. In the case where there are multiple accounts, the user must choose which account. The following image shows an example of the *Account Reset Completion* page with multiple accounts currently restricted:



All user names for all accounts which use the email address are presented in a radio button list. The user must select one account and click the *Next* button. In the example above, selecting john.doe and clicking *Next* would present the user with the following:



The rest of the process is the same as defined above for the single account case. The only difference is that when the password is saved, the restrictions on all accounts that use the same email address are removed.

### Device

An MVC component that provides the ability for a Service Administrator to view, create and modify devices.

Device is also a data object and table in the database. Each device record represents a ClearView device and is attached to a location. Several other data tables are attached to the Device table. These are used for tracking events that occur on devices, device exceptions, state changes and messages to be sent to devices. All scan/treatment records are also attached to a device indirectly through the Calibration table.

#### Purpose

The Device MVC component exists to provide the ability for a Service Administrator to view and administer devices. A Service Administrator is able to view the devices currently defined and make modifications to their configuration. A Service Administrator is also able to add new devices as necessary.

When adding a new device, a Service Administrator must assign the device to a single location. The device’s serial number and revision level are also required to be entered. The serial number is used during the registration of the device by an end-user at a remote site. So it is very important that it be entered correctly. *[Reqs: 4.1.3.3, 4.1.3.4, 4.1.3.5]*

A Service Administrator may also set the number of scan available to the device. This can be done when adding a new device or modifying an existing device. When changing the number of scans available for an existing device, the user must provide “purchase notes” which provide a reason for the change. When the number of scans is modified, EPIC Central will create a tracking record when the device record is updated to track the reason for the change. *[Reqs: 4.1.3.6, 4.1.3.7, 4.1.3.8]*

When a device is added, a unique identifier for the device will be generated. This ID is guaranteed to be unique across the ecosystem. A UID Qualifier is also generated and assigned to the device. This qualifier is provided to the device and must be prefixed to all unique identifiers created by the device to guarantee uniqueness across the ecosystem. *[Reqs: 4.7.2.1, 4.7.2.2]*

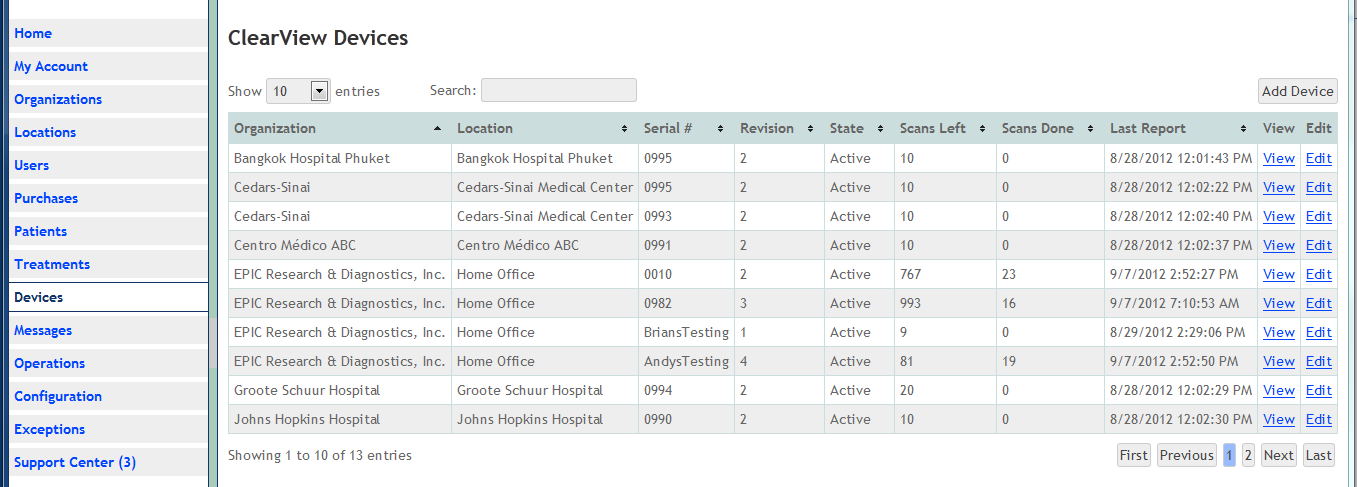
All this functionality is available only to users in the Service Administrator role. No other users can administer devices. *[Req: 4.1.3.2]*

#### Function

This component provides all the necessary functionality to view and administer devices. These functions include the following:

##### List of Existing Devices

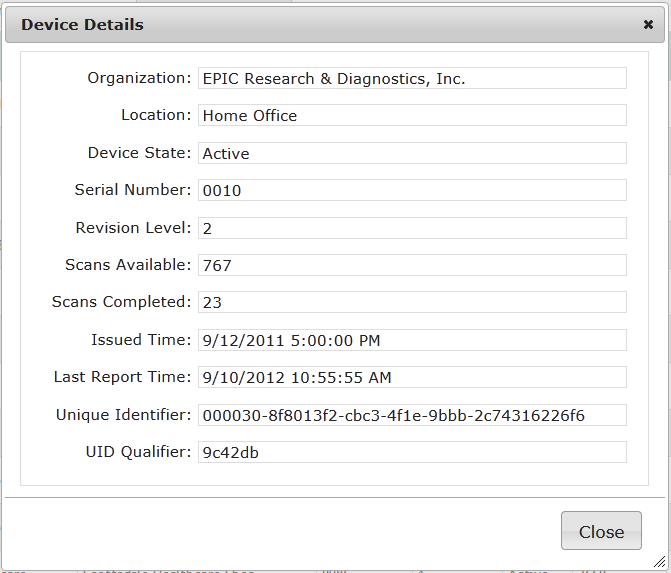
When a Service Administrator clicks the *Devices* entry in the main menu, a list of all currently configured devices is presented. This list contains the devices for all organizations and locations. The organization and location for each device is identified in a column in the list. The list is initially sorted alphabetically by organization. The following image presents an example:



The list only provides the most useful details of each device. A *View* link is provided so the user can see all the details of a particular device by clicking the link. An *Edit* link is also provided to allow modification of a device. The *Add Device* button in the upper right corner allows the user to add a new device.

##### View Device Details

When a user clicks the *View* link for a device in the list of devices, all the details for the configuration of the device will be displayed. The following image presents an example:

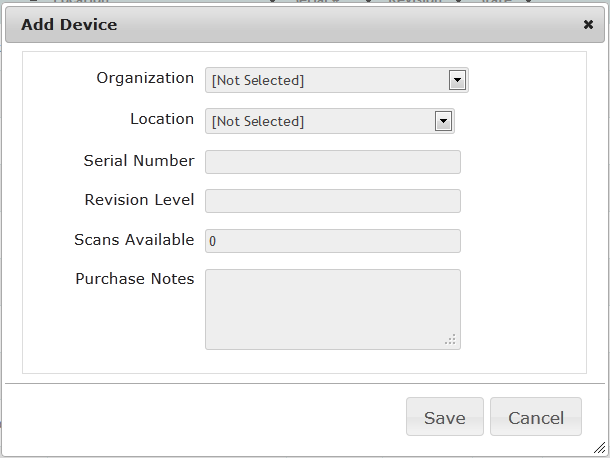


Here’s a short summary of the fields:

* *Organization* – The organization the device belongs to.
* *Location* – The location where the device is deployed.
* *Device State* – The current state of the device. Valid states are *New*, *Active*, *Transitioning*, *Locked* and *Retired*.
* *Serial Number* – The device’s serial number; established at manufacture time and set when the device is added and initially configured.
* *Revision Level* – The device’s hardware/firmware revision level; set when the device is added and initially configured.
* *Scans Available* – The number of scan the device can perform before more must be purchased or transferred to it. A device with no scan available cannot perform a scan.
* *Scans Completed* – The total number of scans performed by the device since it was put into service. This counter cannot be reset.
* *Issued Time* – The date and time when the device was initially added to EPIC Central.
* *Last Report Time* – The date and time when the device last contacted EPIC Central.
* *Unique Identifier* – The universally unique ID for the device.
* *UID Qualifier* – The prefix the device uses for all unique IDs it generates. This hex value is generated when the device is added. It is guaranteed unique among all devices.

##### Add a New Device

When a user clicks the *Add Device* button presented by the table that lists the devices, the following pop-up dialog is displayed:



The *Organization* must be selected before the *Location* can be selected. When the organization is selected the list of locations will be updated with the locations for that organization. When the organization is changed, the list of locations will again be updated.

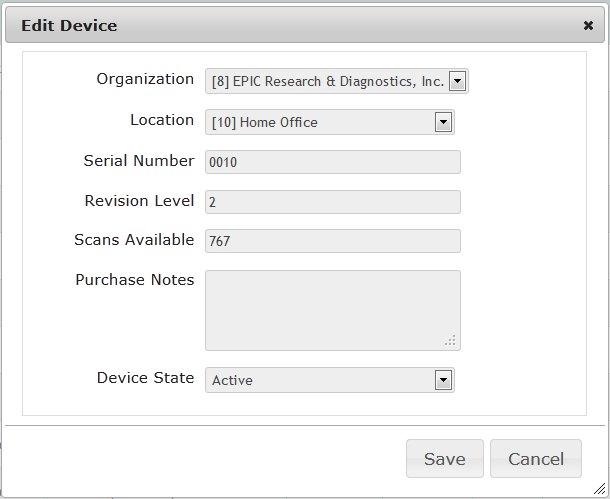
The user must enter the *Serial Number* and *Revision Level* for the device. The user is able to set an initial number of *Scans Available*. If this value is changed, the user must enter a reason in the *Purchase Notes* box.

When all input is complete, clicking the *Save* button will cause the device to be created and the dialog box removed. Clicking the *Cancel* button will remove the dialog box and return to the list of devices with no modifications made.

When the *Save* button is clicked, EPIC Central will validate an *Organization* and *Location* have been selected and the *Serial Number* and *Revision Level* set. If any of these fields are not set, it will display an error message and allow the user to make corrections. If the fields are valid, a unique identifier and UID qualifier will be generated for the device, its state will be set to *New*, its *Issued Time* set to the current time, and it will be saved.

##### Edit an Existing Device

When a user clicks the *Edit* link for a device in the list of devices, a pop-up dialog is presented with the current device configuration and the user can make changes. The following image presents an example:



While the *Device State* is *New*, the user can change the *Organization*, *Serial Number* and *Revision Level*. After a device has been registered and becomes *Active*, none of these values can be changed. The *Location* of a device can be changed; this will force it to the *Transitioning* state.

The user can change the number of *Scans Available*. When doing so, the user must provide a reason for the change in the *Purchase Notes* box.

The user can set the *Device State* to any of the following values:

* *Active* – the device is fully operational
* *Transitioning* – the device is temporarily non-operational
* *Locked* – the device is locked and cannot scan
* *Retired* – the device is permanently taken out of service

When all changes are complete, clicking the *Save* button will cause the device record to be updated and the dialog box removed. Clicking the *Cancel* button at any time will remove the dialog box and return to the list of devices with no modifications made.

When the *Save* button is clicked, EPIC Central will validate that all changes made are valid. If not, it will display an error message and allow the user to make corrections.

#### Data

The Device table in the database contains a record for each device defined in EPIC Central. When a new device is created, another record is added to the table. When an existing device is modified, the existing record is updated.

No records are ever deleted from the Device table. Changing the device’s state to *Retired* effectively removes it from the system. A retired device cannot authenticate to perform any function, nor can it be made active again. *[Req: 4.7.3.1]*

### Message

Device messages can be created and sent to a remote ClearView system from EPIC Central. The *Messages* interface has been designed similar to the *Support Center* and sending an e-mail.

#### Purpose

Messages can be created and sent to ClearView devices and displayed within the ClearView application.

A Service Administrator can view all of the pending messages for a device. *[Req: 4.4.2.7]*

A Service Administrator can create messages and send them to any device, location, organization. When a location or organization is selected as the receiver, the message is assigned to all of the devices within the location or organization. *[Reqs: 4.4.2.1, 4.4.2.2., 4.4.2.3]*

Messages can be edited before they have been received by the device. If a message has been received by some devices, but not by others, the message can be edited and only the devices that have not received the message will receive the edited message. *[Reqs: 4.4.2.5, 4.4.2.4]*

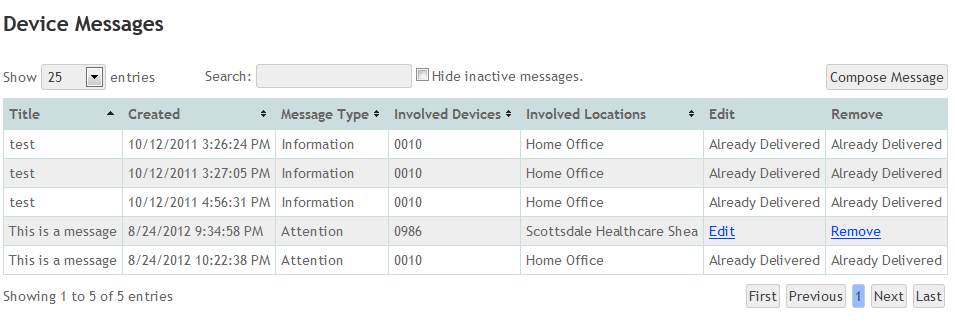
A message can be removed before it is received by the device. *[Req: 4.4.2.6]*

#### Functions

The *Message* page provides all of the functionality to manage messages to be sent to devices. These functions are only available a Service Administrator and include the following:

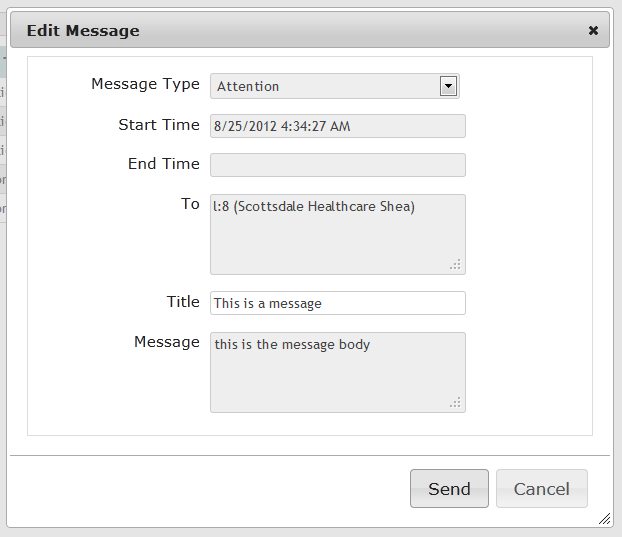
##### List of Messages

The list of messages shows all messages sent to devices as well as any pending messages that have not been received and are available for editing. A sample of this list is displayed below:



##### Editing a Message

A message that has not been received can be edited. By clicking edit, the edit dialog appears. The *To* field is a list of devices, locations or organizations that the message should be sent to. If a device has already received a message, the device will be removed from the list, and the editing will only apply to the new list of devices. If all the devices are listed of a larger denomination such as a location or all of the devices within an organization, the contents of the *To* field will display the largest entity that describes the list of devices.



The *Start Time* and *End Time* define the time range that a message should be displayed. The *Message Type* can affect how the message is displayed within ClearView.

The *Title* and *Message* fields are the primary focus of what is displayed within the ClearView application.

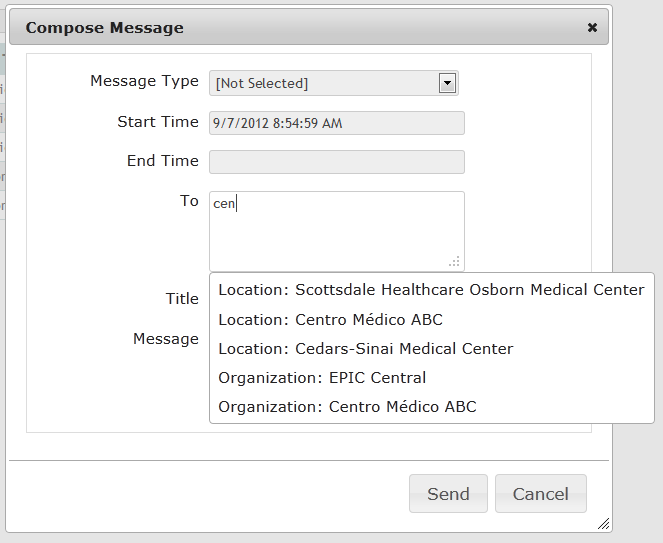
##### Removing a Message

Clicking the *Remove* link next to a message will make the message inactive and hidden. This will prevent the message from being received by any more devices.

##### Adding a New Message

Similar to the edit dialog, a new entry will be created in the *Messages* list when the add dialog is completed and saved. Devices can be added to receive the message by clicking *Edit* and typing in new devices, locations, or organizations to include in the message.

Devices are listed when characters are entered in the *To* field. This feature is shown below:



### System Settings

EPIC Central has a few configuration options that impact its overall behavior. These configuration settings can be modified on the *Configuration* page.

#### Purpose

The *Configuration* page allows quick and easy access to a few of the EPIC Central settings, such as disabled device access to EPIC Central. *[Req: 4.1.6.1]*

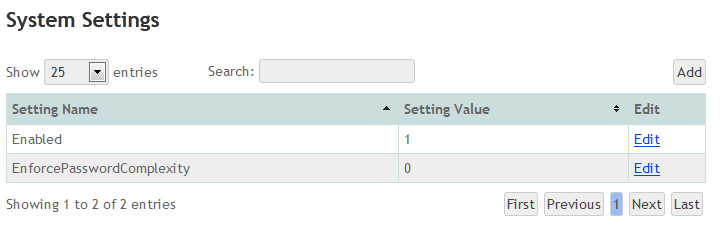
This page is only accessible by a Service Administrator. *[Req: 4.1.6.1]*

#### Functions

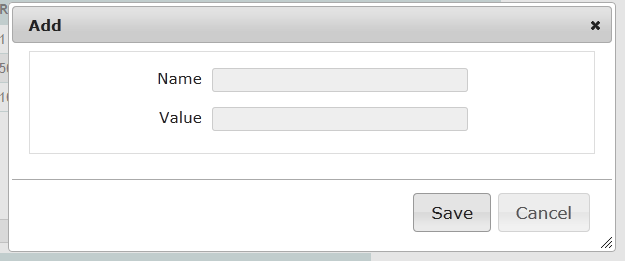
The *Configuration* page lists the current settings.

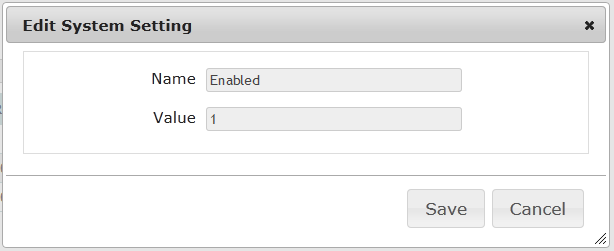
##### System Settings

System configuration settings can be edited or added using the edit buttons next to current settings or the add button above the list of settings.



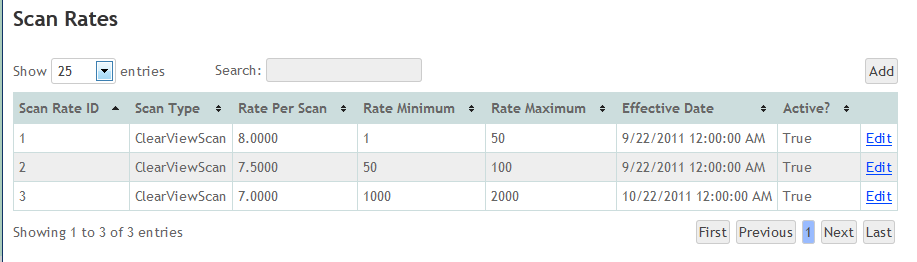
##### Editing System Settings





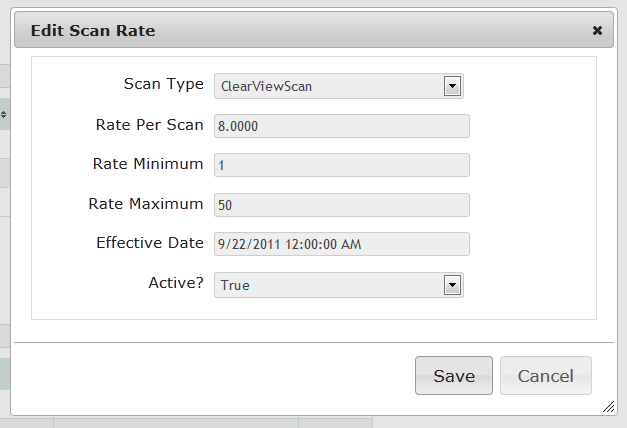
##### Rate Settings

The *Configuration* page also provides an interface for changing the scan rates.



##### Edit a Rate

To edit a rate, the *Scan Type* must be selected, the *Rate Per Scan* field is the item price, the *Rate Minimum* and *Rate Maximum* is the range of scans that must be purchased for the price to be applied. The *Effective Date* is when the rate becomes effective.



## ClearView Cloud

EPIC Central is the receiver of data from ClearView client devices. Devices send their data to be viewed on EPIC Central in a centralized location across entire organizations. This process has a few important parts, the web services described in section *6.7, Web Services*, the Synchronizer Service references in the references section of this document, and the displays for the data described below.

### Purpose

The purpose of the ClearView cloud is to provide a similar interface to ClearView on the web where all patient data can be viewed just as it is on the ClearView device.

A Service Administrator can view all patient records, other users can only view the patients within their organization. *[Req: 4.2.3.3]*

The *Patients* page lists patients uploaded from ClearView. *[Req: 4.2.3.1]*

Patients can be searched by first name, last name, gender, and date of birth. *[Req: 4.2.3.2]*

The *Treatments* page lists all of the treatments for the Service Administrator, and an organization’s treatments for other users. *[Reqs: 4.2.4.1, 4.2.4.3]*

Treatments are searchable by first name, last name, treatment type, treatment time, and location. *[Req: 4.2.4.2]*

The treatment display is modeled after the ClearView treatment display. *[Req: 4.2.4.4]*

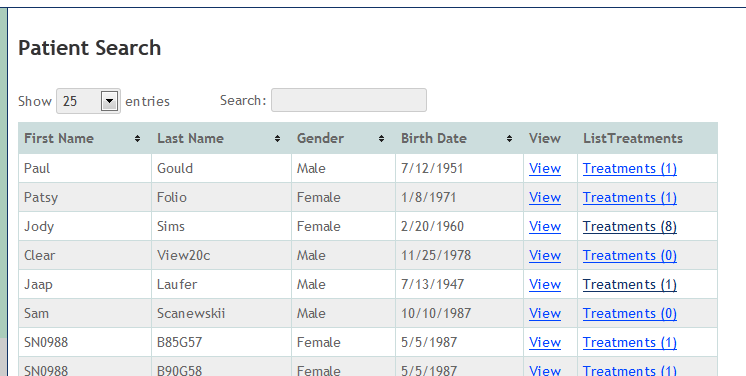
### Functions

The ClearView cloud has two primary functions, viewing uploaded patients, and viewing patient treatments.

#### Patient Search

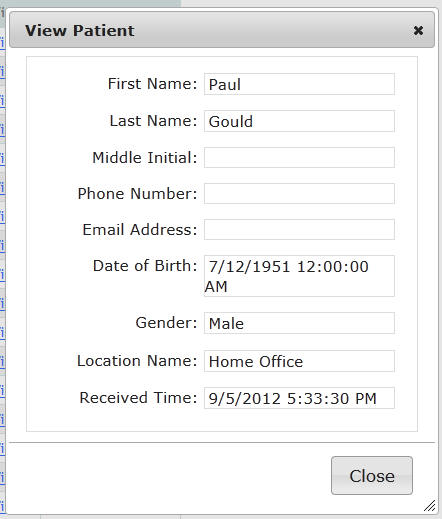
A list of patients can be searched. A Service Administrator has access to view all patients. An Organization administrator can only view patients uploaded from devices within their organization. A regular user can only view patients within their assigned locations.

The list below shows all searchable columns and some of the patient information.



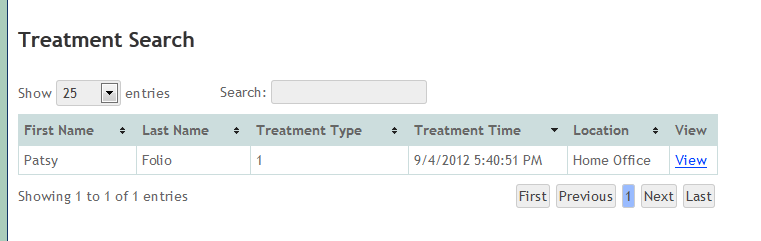
#### Viewing Patient Details

The entire patient record can be viewed using the *View* link for a patient.



#### Listing Patient-Treatments

The treatments for a particular patient are displayed when the treatments link is clicked next to the patient name.



#### Treatment Search

By default the treatments are sorted by time. All columns are searchable.



A treatment is viewable using the *View* link on the right.

#### Viewing a Treatment

The treatment display is modeled after the display in ClearView to provide a consistent experience for all users of the system.



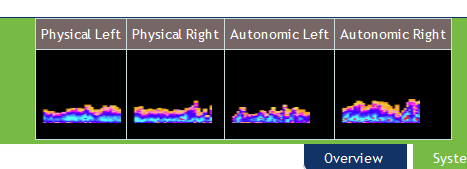
A treatment has a few primary functions displayed across the top right, these are described in detail below.

##### Overview

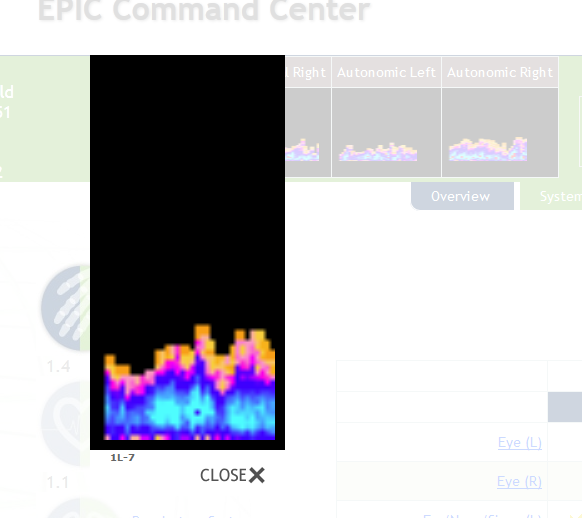
The treatment overview is a summary of the scan data. Clicking on an organ system will display the results for each component of that organ system. The organ system is illustrated on the left, and the menu button is selected.



Clicking on an organ component will bring up the relevant image sectors in the top display.

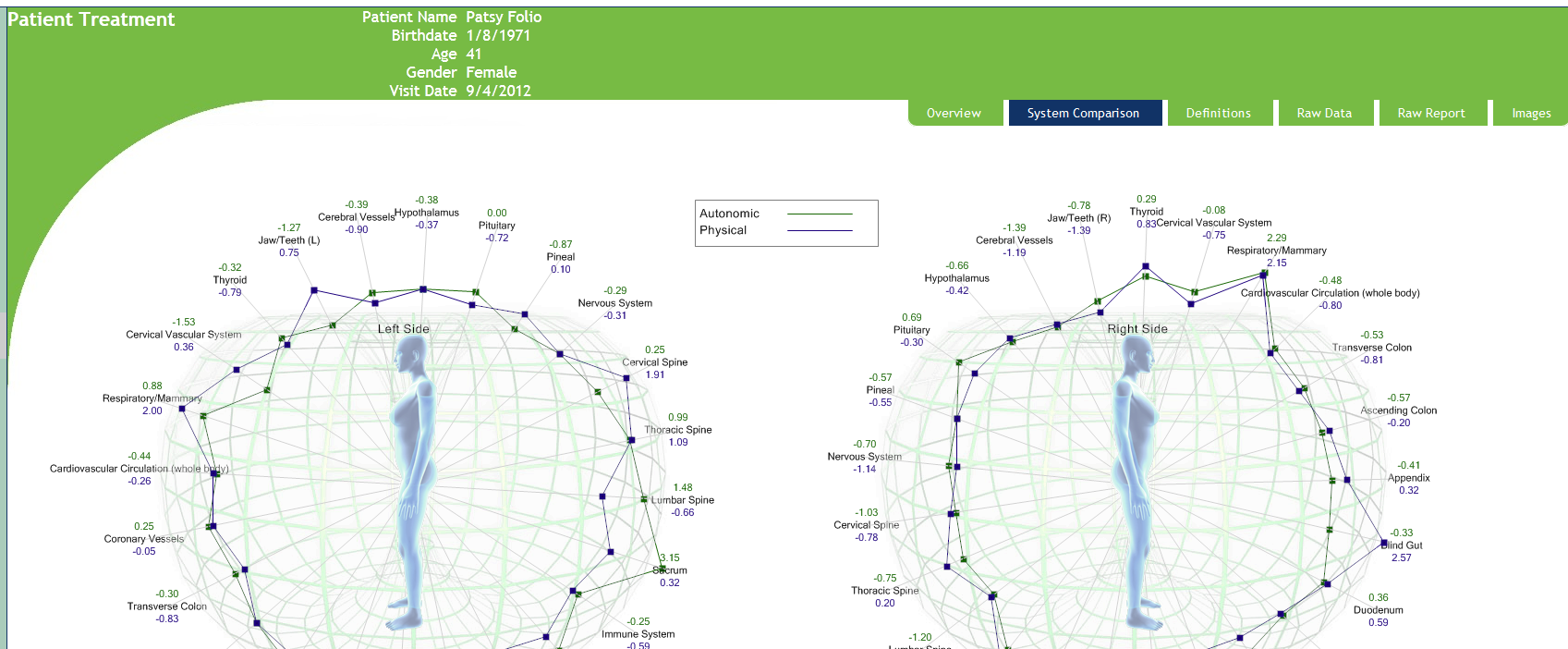


Clicking on an image sector will display the image and the mouse wheel can be used to scroll in to view the image larger.



##### System Comparison

The system comparison page functions just like ClearView. It shows a comparison between left and right sides of the body with specific points.



##### Raw Data

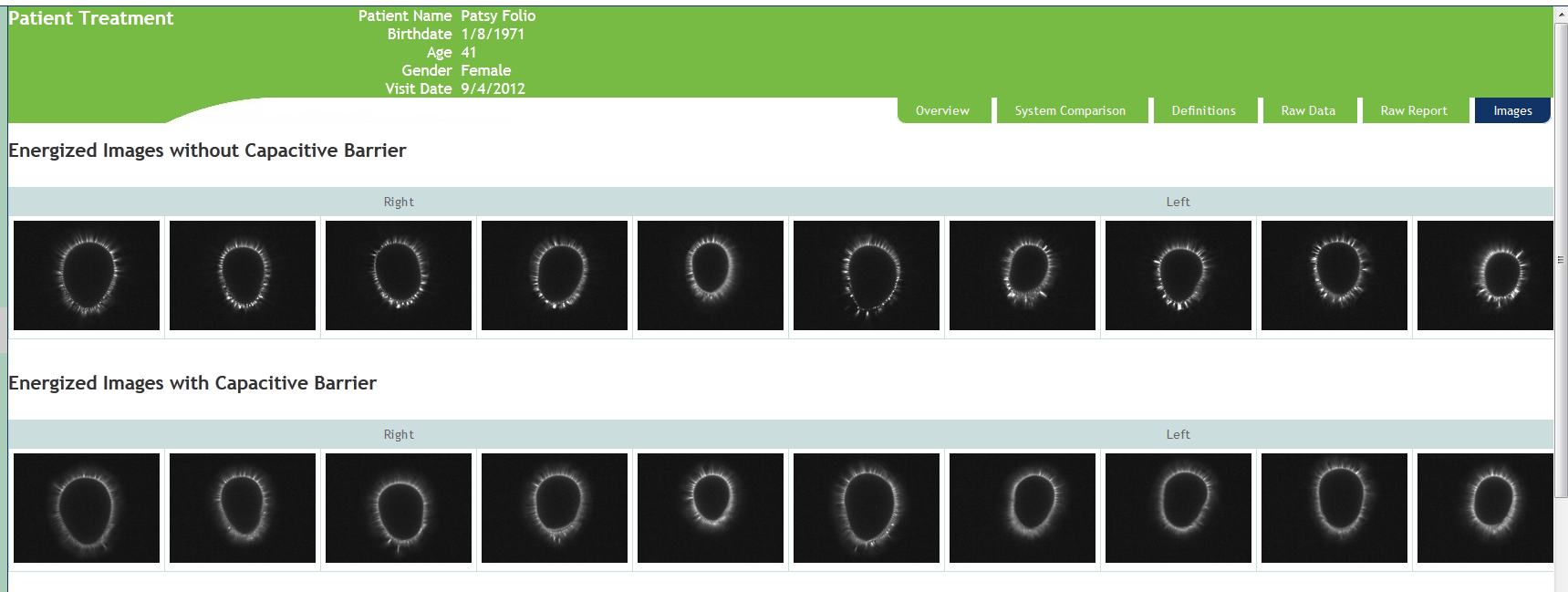
The Raw Data page shows the Analysis Results table in full for the specified treatment. This tab is only available to a Service Administrator.

##### Raw Report

The Raw Report page displays the NB Analysis table and the Calculation debug table for the specified treatment. This tab is only available to a Service Administrator.

##### Images

Shows the patient images from which the data is generated. This tab is only available to a Service Administrator.



## ClearView Earth

The ClearView Earth page shows a Google Map of all the organizations and locations and their current activity.

### Purpose

A location can be selected and all the device’s activities are shown. *[Req: 4.2.2.8]*

Icons on the map at the locations coordinates indicate the most recent activity for any of the devices at the location. *[Reqs: 4.2.2.4, 4.2.2.5]*

Statistics about the overall system are displayed above the map. These values are all refreshed every couple of seconds. *[Reqs: 4.2.2.2, 4.2.2.3]*

The *Operations* page is only available to the Service Administrator. *[Req: 4.2.2.1]*

The map can be clicked and dragged to view different areas of the world. *[Req: 4.2.2.6]*

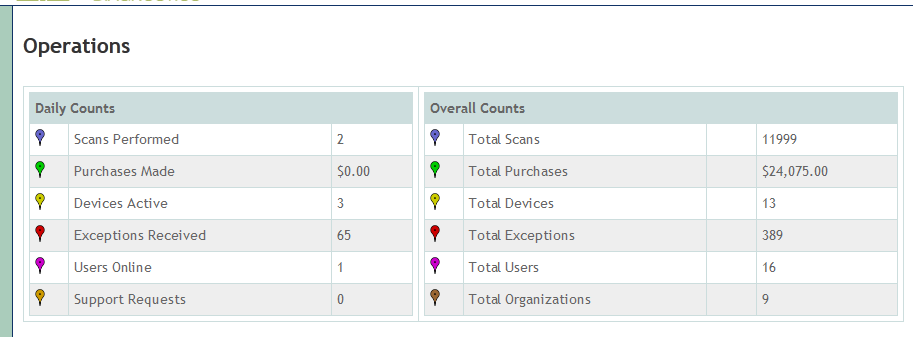
Controls over the map can be used to zoom in and out. *[Req: 4.2.2.7]*

### Functions

The *Operations* page shows all the necessary information. This menu entry is only available to a Service Administrator.

#### Statistics

The statistics portion of the map displays two columns of data; the right hand column indicates when values have gone up using a green up arrow.



The statistics are refreshed based on certain actions that occur throughout EPIC Central or every couple of seconds if no other action/event occurs. This section is like a key to the map displayed below it.

#### Map

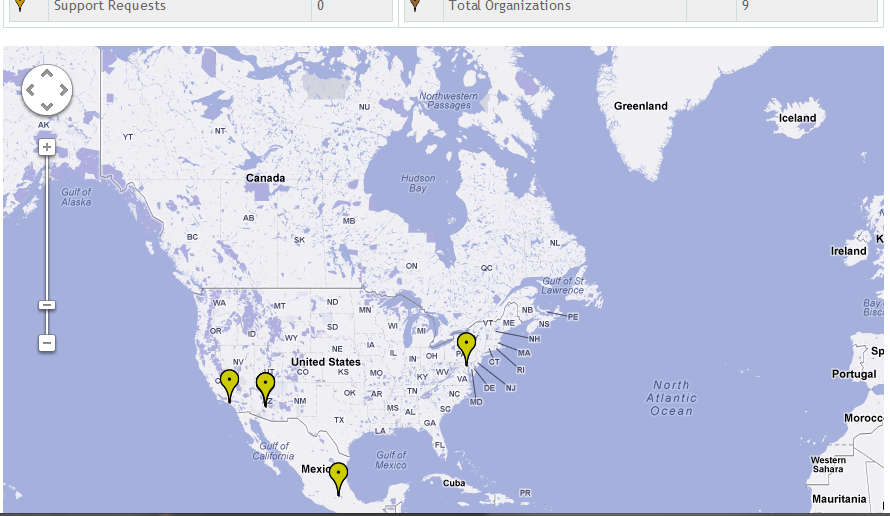
##### Navigational Controls

The map functions as a normal Google Map, with navigational controls in the upper-left corner.



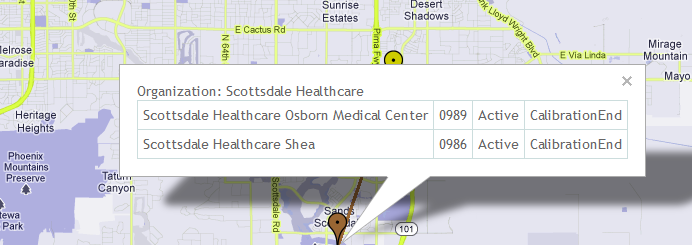
##### Map Markers

All locations are shown on the map and ordered by significance of the event or entity type. Organizations are displayed behind the location icons. When an event occurs, the icon is displayed on top.



##### Location Information

The icons on the map are interactive and when clicked they display a dialog with device information for that location.



## Purchases

Users can make purchases for any device at any location they have access to. There are a few main functions relating to purchases and the checkout process.

### Purchase History

The primary function of the *Purchases* page is to display a list of purchases, descriptions of the purchases, user who made the purchase and date and time.

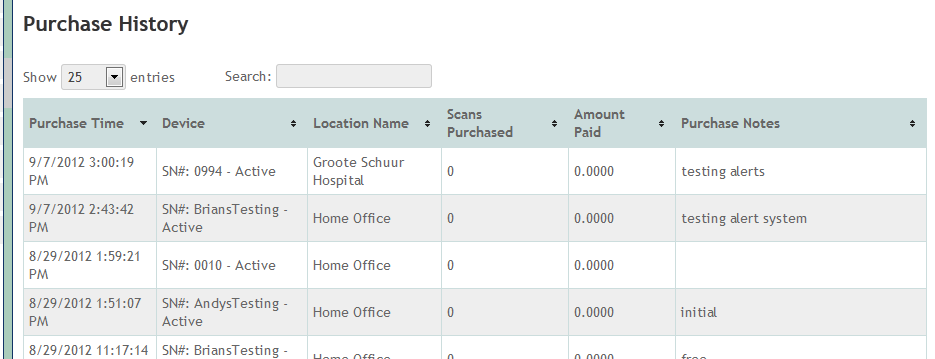
#### Purpose

The *Purchase History* section shows all transactions, including when the number of scans available is changed by a Service Administrator. *[Req: 4.1.3.8]*

Transactions are searchable by all required fields displayed in the table below. *[Req: 4.2.5.2.6]*

#### Functions

A sample of the *Purchase History* is displayed below; all columns shown are searchable by the user:



The top two results are logged from modifying the scans available field on the device directly.

### Making a New Purchase

Creating a purchase is similar to the shopping card model. A user selects the type of scan they want, they enter the quantity between a given range configured in the Rates table, and finally, the user checks out and the transaction is authorized.

#### Purpose

Users can purchase scans at the device level. A Service Administrator can purchase scans for any device, an organization administrator can purchase scans for any device in their entire organization, and a regular user can purchase scans for devices in their assigned locations.

Authorize.Net is the payment gateway used. *[Req: 4.2.5.1.1]*

All credit card information is stored in the Autorize.Net Customer Information Management (CIM) system. No crucial credit card data is stored on EPIC Central. *[Req: 4.2.5.1.2]*

A user must select the device to purchase scans for; the user only has access to devices at locations that they are assigned to. *[Req: 4.2.5.2.1]*

Organization administrators can make purchases for any device within their organization. *[Req: 4.2.5.2.2]*

When a purchase is completed, the available scan count for each device involved in the purchase is incremented according. *[Req: 4.2.5.2.3]*

All transactions for each device are logged. *[Req: 4.2.5.2.5]*

#### Functions

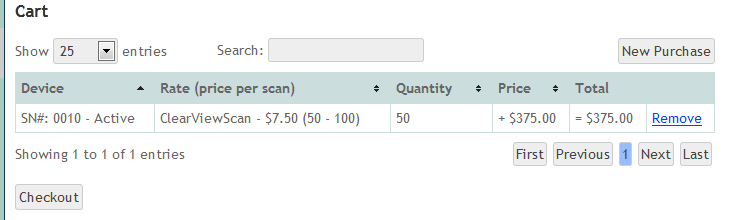
This page provides all the functionality to make a new purchase.

##### The Cart

The cart will be populated with purchase items.



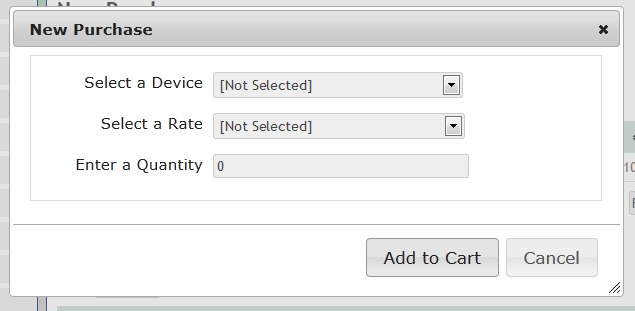
The cart is populated with the new purchase and the user can click *Checkout* or add another device purchase.



To create a new purchase, the user must click the *New Purchase* button on the *Purchases* page.

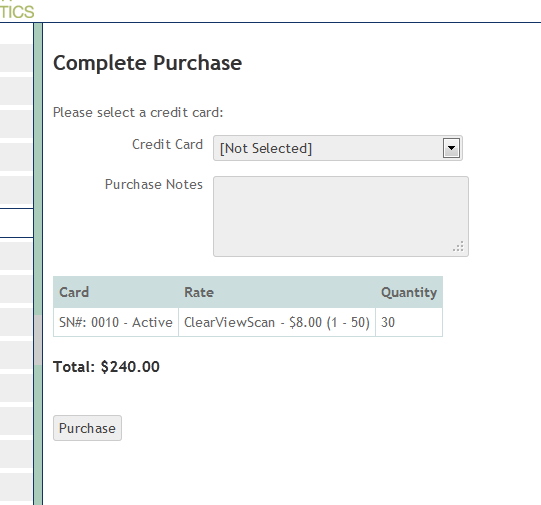
##### New Purchase Dialog

The device must be selected from the list of accessible devices. The rate is selected from the list of rates configured by a Service Administrator. The quantity must be within the range of the rate.



##### Checkout Page

The final step in the purchase process is the *Complete Purchase* page, where the user selects the card to use, confirms the purchase list and total, and clicks *Purchase*.



### Managing Credit Cards

The *New Purchase* page includes a section for managing a user’s credit cards.

#### Purpose

A user can add a credit card to their user account for use during the checkout process. Credit cards are stored on a per user basis.

A user must enter all of the credit card information in order to use it for new purchases. This includes the CCV code. *[Req: 4.2.5.1.5]*

After the user enters the credit card, EPIC Central stores a small subset of the information just for reference. This display is shown in an image below. *[Reqs: 4.2.5.1.3, 4.2.5.1.4]*

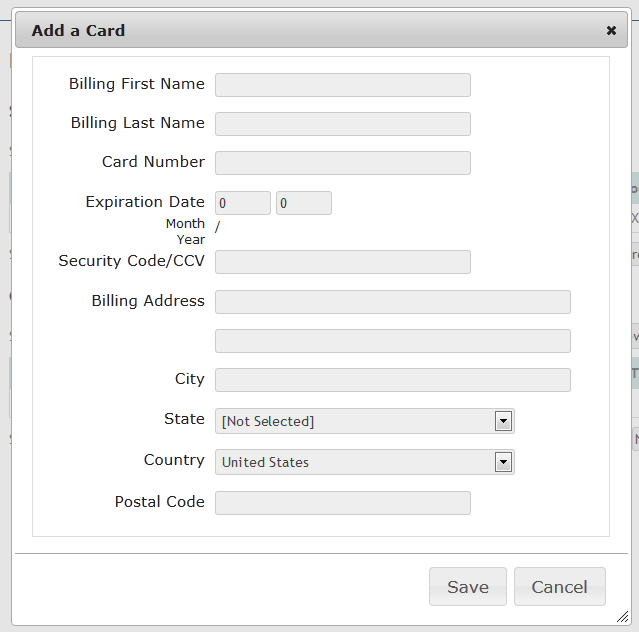
After the card is saved, the card can be selected on the checkout screen by the user who added it which allows it to be used to authorize a new purchase. *[Req: 4.2.5.1.6]*

#### Functions

The *New Purchase* page provides all the functionality for storing and editing credit card information.

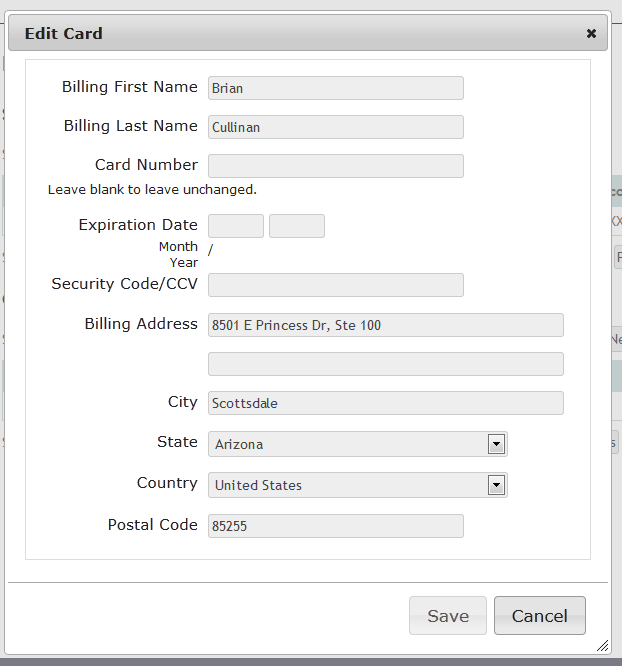
##### Adding a New Card

The user selects the *Add Card* button and a dialog appears. All the necessary information is collected.



##### Editing an Existing Card

The user can also select the *Edit Card* where the card number and CCV code can be left blank to leave as is and only modifying the billing address.



### Transferring Scans

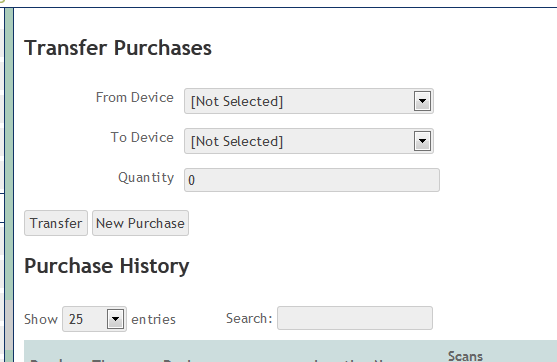
Scans are transferrable between devices.

#### Purpose

A simple form must be completed and submitted and the scans available will be deducted from one device and added to the other. Both events are recorded in the transaction log. The *From Device* and *To Device* fields must be different from each other. *[Req: 4.2.5.2.4]*

#### Functions

The user selects the device to transfer scans from, the device to transfer scans to, and the *Quantity* between 1 and the number of scans available to the *From Device*. The activity is logged by this submission.



## Support Center

*Support Center* is an area designed for users to reach a System Administrator or other user from within EPIC Central.

### Purpose

*Support Center* is designed to look and act similarly to any e-mail client such as Gmail or Outlook.

*Support Center* is only available to authenticated users. *[Req: 4.4.1.1]*

A support message can be created and sent to any user within the logged in user’s organization, or a System Administrator. *[Req: 4.4.1.2]*

A support message requires at least the following to be filled in: message body, subject and to/recipient. *[Req: 4.4.1.3]*

A user can view the message and use the *Reply* button to reply with another message. *[Req: 4.4.1.4]*

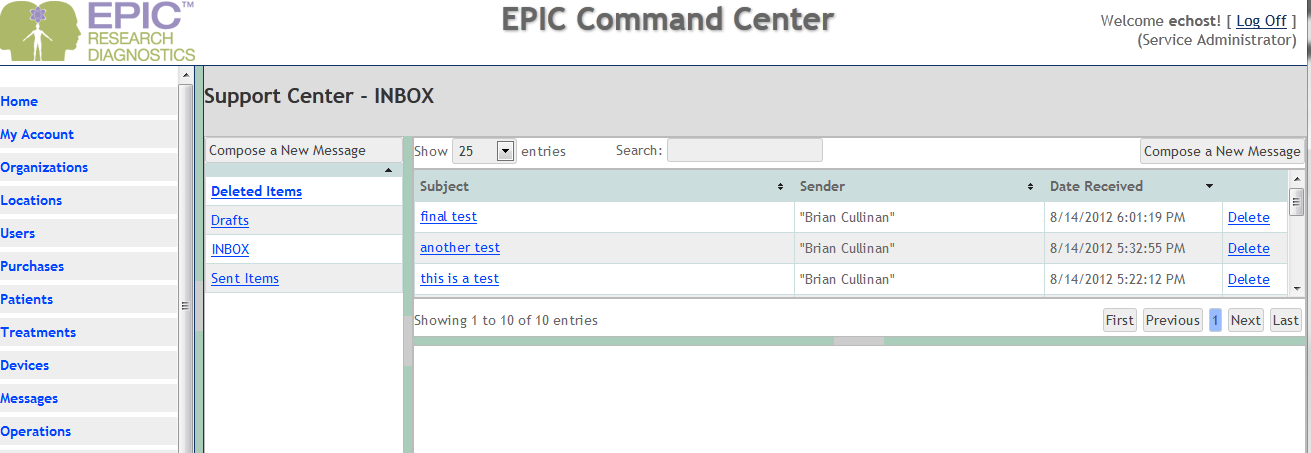
All messages can be viewed on the *Support Center* default view, similar to an Outlook Inbox. *[Req: 4.4.1.5]*

A *Delete* link is provided next to each message for hiding messages from being displayed. *[Req: 4.4.1.6]*

Every few minutes a service runs to check the *Inbox* count. When the count changes the *Operations* view will reflect the change. *[Req: 4.4.1.7]*

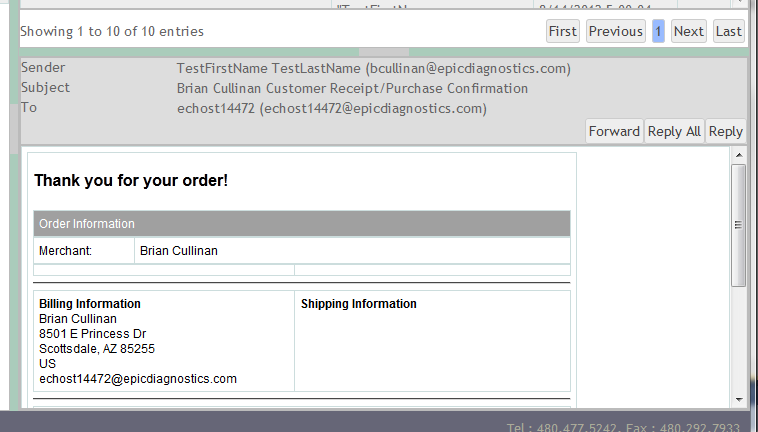
### Messages

The primary view for *Support Center* is a list of folders and received messages.

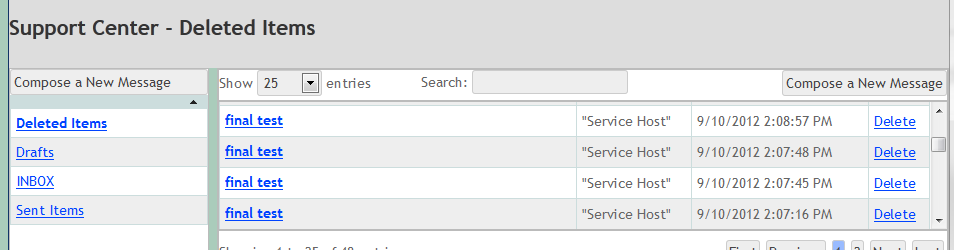


The *Delete* link can be seen in the image above.

When a message is clicked it is loaded in the bottom portion of the window.

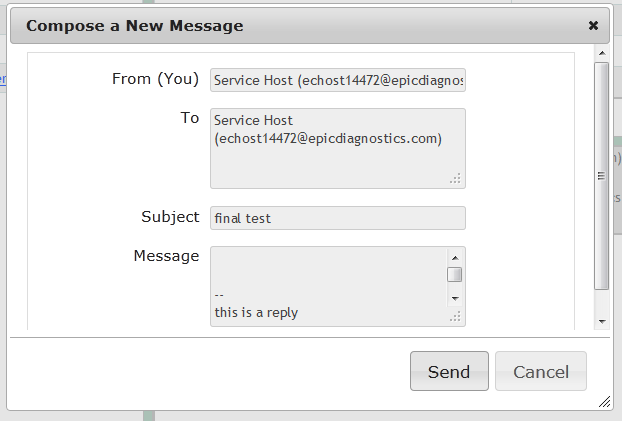


Clicking the folder on the left hand side will change the location of the mailbox being shown. For example, a user can view a list of their deleted messages, with the option to permanently delete any message:

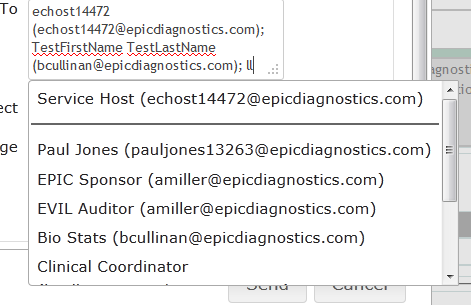


### Composing

A user can compose new messages, edit their drafts, and reply to a message all using the same interface.



When text is typed in to the *To* field, the list of available recipients is shown for auto-completion:



When a text field loses focus, i.e. the cursor is moved away from the field, the message is automatically saved in *Drafts*. This behavior is indicated by a “Saving…” message and the *Send* and *Cancel* buttons are temporarily disabled.

## Web Services

EPIC Central includes a set of web services that allow a ClearView device to interact with EPIC Central in a highly secure manner with almost no manual intervention. A human user must be involved during the device registration process. But after a device is registered, it can authenticate itself and interact in a complete automated manner.

The web services are designed using the Representational State Transfer (REST) paradigm on top of the Windows Communications Framework (WCF). REST is a stateless paradigm; no state is maintained by EPIC Central. Each web service invocation is an atomic operation and is dependent on no other operation.

The design of the web services allows for running multiple versions of the service package simultaneously. This is necessary since it will be impossible to update all ClearView client systems at the same time. As the web services evolved we must be able to implement new functionality in a way that may not be supported by clients running older versions. However, if EPIC Central can run multiple versions at the same time, older clients can continue to use the older versions of the web services while new clients use the latest version.

REST services are lightweight and use simple HTTP status codes to communicate success or failure. This concept is a key part of the EPIC Central web services. When a service invocation completes successfully a “200 OK” HTTP status code is returned by the WCF. If any error occurs, an exception is thrown which the WCF will convert to the HTTP status code specified by the exception. The EPIC Central services are designed to produce an error subcode that gives additional information about an error. These subcodes are returned as strings in a standard format so they can be easily parsed by the client. The following table lists all the errors returned by the services. The ID for each error is used in the following sections to reference rows in this table. It is not part of the error response to the client.

|  |  |  |
| --- | --- | --- |
| **ID** | **Error Subcode** | **HTTP Status Code** |
| 401.1 |  | 401 Unauthorized |
| 401.2 | (401) Device authentication required | 403 Forbidden |
| 402 | (402) Basic user authentication required | 403 Forbidden |
| 405.1 | (405) Location not found | 404 Not Found |
| 405.2 | (405) Location not found | 412 Precondition Failed |
| 406.1 | (406) Device not found | 404 Not Found |
| 406.2 | (406) Device not found | 412 Precondition Failed |
| 407.1 | (407) Patient not found | 404 Not Found |
| 407.2 | (407) Patient not found | 412 Precondition Failed |
| 408 | (408) Calibration not found | 412 Precondition Failed |
| 409 | (409) Image set not found | 412 Precondition Failed |
| 410.1 | (410) Organization invalid | 412 Precondition Failed |
| 410.2 | (410) Organization invalid | 406 Not Acceptable |
| 411.1 | (411) Location invalid | 412 Precondition Failed |
| 411.2 | (411) Location invalid | 406 Not Acceptable |
| 412 | (412) Device invalid | 412 Precondition Failed |
| 414 | (414) Treatment invalid | 409 Conflict |
| 415 | (415) Calibration invalid | 412 Precondition Failed |
| 416 | (416) Image set invalid | 409 Conflict |
| 420 | (420) Device state invalid | 406 Not Acceptable |
| 421 | (421) Device has no scans | 406 Not Acceptable |
| 430 | (430) GUID invalid | 412 Precondition Failed |
| 431 | (431) Checksum invalid | 400 Bad Request |
| 501 | (501) Internal database error | 500 Internal Server Error |
| 502 | (502) Internal I/O error | 500 Internal Server Error |
| 503 | (503) Internal transaction error | 500 Internal Server Error |

A “service API” package is part of the EPIC Central web services implementation. This package contains the classes that define the common interface shared between the client and EPIC Central. All the objects that are passed as parameters and the definitions for constants and error code are contained within the service API package. It also contains client side implementations of the various web service methods. These client implementations contain all the lower-level plumbing necessary to serialize output, invoke web services, and deserialize responses and errors. This package provides a consistent implementation for multiple clients, although at the present, the ClearView Synchronizer is the only client.

The following sections define the EPIC Central web services.

### Authentication

Authentication is not a web service, but a process required by every web service invocation to ensure that only authorized users and devices can access each web service.

#### Purpose

Authentication ensures that the web services exposed are accessed only by authorized persons or devices. If authentication fails, access to the requested service is always denied. *[Req: 4.6.2]*

#### Function

Web service authentication is performed by a custom .NET HTTP module which intercepts every HTTP request coming to the EPIC Central site. It effectively ignores any request that is not destined for a web service and allows the request to continue. If the request is for one of the web service methods, it will authenticate the request before allowing it to continue. If it cannot be authenticated, a “401 Unauthorized” HTTP status code is returned.

Authentication is implemented using HTTP basic authentication, which uses simple user name/password credentials. Since all access to EPIC Central is limited to HTTPS, credentials are always sent over an encrypted connection, never in the clear.

Some services require authentication of a user. These use the standard user name and password for credentials. The user will enter these credentials when prompted by the ClearView application and the application will simply pass them through to EPIC Central as part of the web service invocation.

Some services require the device to authenticate. This allows the device to run a service that is not necessarily associated with an EPIC Central user. Device authentication mimicks the standard user name and passwords credentials except the user name is actually the device’s unique identifier and the password is a random token. Both of these values are generated when the device registers with EPIC Central and are provided to the device in the registration response. The device simply saves them and uses them when necessary to invoke a service. *[Reqs: 4.6.4, 4.6.5]*

#### Inputs

When user authentication is required an EPIC Central user’s user name and password are required. When device authentication is required a device’s unique identifier and authentication token are required.

#### Outputs

There are no direct outputs. If authentication is successful, the web service is invoked. If it isn’t successful, an error is produced.

#### Errors

If authentication fails, a “401 Unauthorized” HTTP status code is returned.

### Registration

ClearView devices are required to register with EPIC Central before they can access any other services. Registration is required for new devices (state is *New*) and devices being moved to another location (state is *Transitioning*). Since a device doing registration is not registered and cannot authenticate itself, the ClearView user performing the registration must authenticate as an EPIC Central user. *[Reqs: 4.3.1.1, 4.3.1.2, 4.3.1.3]*

Prior to performing device registration, the device must be added to EPIC Central by a Service Administrator; see section *6.2.5.2.3, Add a New Device*. During registration, the ClearView user must select the location for the device and enter the device’s serial number. These must match the settings the administrator used when adding the device. *[Reqs: 4.3.1.4, 4.3.1.5, 4.3.1.6]*

When a device has completed registration, its state will be set to *Active* and it will be fully operational. *[Req: 4.3.1.7]*

The methods described here are all part of the *RegistrationService* class. All these methods allow user authentication as an EPIC Central user because until the device is registered it cannot authenticate itself. *[Req: 4.6.3]*

#### Method: GetLocations

##### Purpose

This method allows a device to get the list of locations for the user’s organization.

##### Function

This method finds all the locations configured for the user’s organization and returns that list to the client. The user is the person who has authenticated as an EPIC Central user through the ClearView UI.

##### Inputs

This method requires no parameters. A user must be authenticated when it is invoked.

##### Outputs

If this method completes successfully, it returns a list of the locations defined for the user’s organization. Each item in the list contains the name of the location and the location’s unique identifier.

##### Errors

The method can produce the following errors returned to the client:

* 402 – If invoked without an authenticated user
* 405.1 – If the user’s organization has no locations configured

#### Method: RegisterDevice

##### Purpose

This method registers a device with EPIC Central so it is fully functional. *[Reqs: 4.6.1.1, 4.6.1.2]*

##### Function

This method locates the device using an input serial number and registers it at the specified input location. After registration, the device state is *Active* allowing it to perform all device functions. This method requires a user to be authenticated since the device is not yet registered when it is invoked.

##### Inputs

This method requires the following input parameters:

* Location UID – the unique identifier for a location from the list of locations retrieved via the GetLocations method
* Serial Number – the serial number for the device

##### Outputs

If this method completes successfully, it returns a *Credentials* object containing the following parameters:

* Organization UID – the unique identifier of the organization the device belongs to
* Location UID – the unique identifier of the location where the device has been registered
* Device UID – the unique identifier for the device itself
* UID Qualifier – a string of six hex digits that the device will prepend to every unique identifier it creates; this is unique across all devices in EPIC Central and guarantees IDs generated by devices are then unique across the entire ecosystem *[Req: 4.7.2.2]*
* Authentication Token – a random authentication token (password) the device must use to authenticate with EPIC Central; EPIC Central does not store the token itself, only a SHA-512 hash of the token

##### Data

If completed successfully, this method will update the state of the device to be *Active* and store the hash of the authentication token in the device’s entry in the *Device* table. If it completes with an error, no data is modified.

##### Errors

This method can produce the following errors returned to the client:

* 402 – If invoked without an authenticated user
* 405.2 – If the location UID provided as a parameter does not match a location in EPIC Central
* 411.1 – If the location is found using the location UID parameter but the location’s organization does not match the user’s organization
* 406.1 – If there is no device defined in EPIC Central with the serial number provided as a parameter
* 420 – If the device is not in a state that allows it to be registered; a device must have either *New* or *Transitioning* state to be registered

#### Method: ResetAuthenticationKey

##### Purpose

This method allows a device’s authentication token to be changed to a new value if there is any security-related need to do so. *[Req: 4.6.1.3]*

##### Function

After fully authenticating the device, this method will generate a new authentication token, store the hashed value of the token in the database, and send the new token to the client device. This method allows authentication by either the device itself (using its current token) or an EPIC Central user using user name and password.

##### Inputs

This method requires the following input parameters:

* Device UID – the device’s unique identifier provided at registration
* Location UID – the unique identifier for the location where the device is deployed
* Organization UID – the unique identifier for the organization that the location is part of

##### Outputs

If this method completes successfully, it returns a string that contains the new authentication token. The next time the device authenticates, it must use the new token.

##### Data

If this method completes successfully, it stores the hash of the new authentication token in the device’s entry of the *Device* table. If it completes with an error, no data is modified.

##### Errors

This method can produce the following errors returned to the client:

* 401.2 – If not authenticated as either device or user
* 406.1 – If authenticated as a user and no device is found using the input device UID parameter
* 406.2 – If authenticated as a user and device is found using the input device UID parameter but the device’s organization does not match the user’s organization
* 412 – If authenticated as a device but the authenticated device’s UID does not match the device UID parameter
* 411.1 – If device’s location’s UID does not match the location UID parameter
* 410.1 – If the device’s location’s organization’s UID does not match the organization UI parameter
* 420 – If the device’s state is not *Active*

### Events

ClearView devices report operational events to EPIC Central where they are stored providing a historical record of operations performed by the devices. These events also update the operations map that allows a Service Administrator to watch activity at any or all locations. *[Req: 4.6.1.13]*

The methods described here are all part of the *EventService* class.

#### Method: Create

##### Purpose

This method allows a device to report a significant event to EPIC Central. The list of events that can be reported is predefined and shared by the device and EPIC Central.

##### Function

This method receives an event for a device, stores the event in the database and updates the current status of the device based on the event.

##### Inputs

This method requires a single parameter, an *Event* object that contains the following fields:

* Event ID – primary key assigned by the database; not used as input
* Type – the type of event being reported; can be one of the following:
* Ping – the device is functioning; also allows the device to know EPIC Central is alive
* Scan Begin – the device has started a scan
* Scan End – the device has finished a scan
* Calibration Begin – the device has started a calibration
* Calibration End – the device has finished a calibration
* Application Begin – the ClearView application has started
* Application End – the ClearView application is shutting down
* Time – date/time when the event occurred on the device; must be UTC value

##### Outputs

If this method completes successfully, it will return the same *Event* object it received as input except it will insert the Event ID generated by the database as the primary key for the event record created.

##### Data

If this method completes successfully, it will create a new entry in the *DeviceEvent* table to record the event and it will update the device’s entry in the *Device* table to reflect the current status (the type of event reported) and last report time.

##### Errors

This method can produce the following errors returned to the client:

* 401.2 – If device is not authenticated

### Scan Authorization and Recording

ClearView devices validate their ability to perform a scan before allowing a scan to start. This validation includes checking that the device is properly registered and that it has scans available. When a scan is complete, a device will report the completion of the scan to EPIC Central. A device can also get the number of scans it has available at any time.

The methods described here are all part of the *ScanService* class.

#### Method: Validate

##### Purpose

This method validates a device’s ability to perform a scan. *[Req: 4.6.1.4]*

##### Function

This method will validate that the device’s stored location and stored organization match the device’s location and organization on EPIC Central, that the device’s state is *Active*, and that the device has at least one scan left in its scan count. If all these conditions are true, the device can perform a scan. Otherwise, it cannot.

##### Inputs

This method takes a single parameter as input, a *ValidationKeys* object that contains the location UID and the organization UID provided to the device when it registered; these must match the device’s location’s UID and the device’s location’s organization’s UID on EPIC Central. The device must authenticate itself which allows EPIC Central to know which device it is.

##### Output

This method returns no output; it does return “200 OK” if successful. If any validation fails, it will produce an error. Thus, if no error is returned, the device is validated to perform a scan.

##### Data

This method updates no data.

##### Errors

This method can produce the following errors returned to the client:

* 401.2 – If device is not authenticated
* 411.2 – If location UID in input *ValidationKeys* does not match device’s location’s UID
* 410.2 – If organization UID in input *ValidationKeys* does not match device’s location’s organization’s UID
* 420 – If device’s state is not *Active*
* 421 – If device’s scan count is zero or less than zero

#### Method: GetCount

##### Purpose

This method allows a device to get the number of scans it has left to perform. *[Req: 4.6.1.5]*

##### Function

This method simply returns the number of scans the authenticated device has available and the number of scans it has completed.

##### Input

This method has no input parameters. It knows the device because it must be authenticated.

##### Output

If successfully completed, this method returns a single output *ScanCount* object that contains the number of scans the device has available and the number of scans it has completed.

##### Data

This method updates no data.

##### Errors

This method can produce the following errors returned to the client:

* 401.2 – If device is not authenticated

#### Method: ScanComplete

##### Purpose

This method allows a device to report the completion of a scan. *[Req: 4.6.1.6]*

##### Function

This method will increment the number of scans a device has completed, decrement the number of scans the device has available, and create a record of the scan completed. *[Req: 4.6.6]*

##### Inputs

This method requires a single input parameter, a *ScanRecord* object that contains the type of scan performed and the timestamp when the scan started.

##### Output

If successfully completed, this method returns a single output *ScanCount* object that contains the number of scans the device has available and the number of scans it has completed.

##### Data

If successfully completed, this method will update the number of scans available and the number of scans completed in the device’s entry in the Device table. It will also add an entry to the *ScanHistory* table to record the scan including the type of scan and the start time of the scan.

##### Errors

This method can produce the following errors returned to the client:

* 401.2 – If device is not authenticated
* 420 – If device’s state is not *Active*
* 503 – If a database error occurs updating the *Device* entry or adding the *ScanHistory* entry

### Data Synchronization

EPIC Central provides the functionality for a user to view the results of a scan from any browser anywhere in the world, provided the user has the proper permissions. To provide this capability all the data associated with patients and scans must be uploaded to EPIC Central and stored. Data synchronization is the process of synchronizing EPIC Central with the data on a ClearView device. There are several separate services for uploading data.

#### Patient Service

Patient records are created and updated on a ClearView device prior to scanning a patient. A patient record contains basic information about a patient. The patient service provides the ability for a device to add patient records to EPIC Central or update existing patient records.

##### Method: Add

###### Purpose

This method allows a device to add a patient to the EPIC Central database. *[Req: 4.6.1.7]*

###### Function

This method receives all pertinent information about a patient from a device and creates an entry in the *Patient* table in the EPIC Central database. If an entry for the patient already exists (found by UID), nothing is changed.

###### Inputs

This method requires a single input parameter, a *Patient* object, containing all information about a patient. The *Patient* object maps directly to the Patient table in the database.

###### Outputs

If this method completes successfully, it returns the same *Patient* object received as input, except it sets the Patient ID to the primary key for the corresponding entry in the Patient table.

###### Data

If this method completes successfully, it adds an entry for the patient to the Patient table in the database.

###### Errors

This method can produce the following errors returned to the client:

* 401.2 – If device is not authenticated
* 420 – If device’s state is not *Active*

##### Method: Update

###### Purpose

This method allows a device to update/change a patient record in the EPIC Central database. *[Req: 4.6.1.8]*

###### Function

This method receives all patient information from a device, looks up the patient record in the database using the UID, and updates the record with the information provided. If the patient record does not exist, an error is returned.

###### Inputs

This method requires a single input parameter, a *Patient* object, containing all information about a patient. The *Patient* object maps directly to the Patient table in the database.

###### Output

This method returns no output; it does return “200 OK” if successful. If any validation fails, it will produce an error.

###### Data

If this method completes successfully, it updates the entry for the patient in the Patient table in the database.

###### Errors

This method can produce the following errors returned to the client:

* 401.2 – If device is not authenticated
* 420 – If device’s state is not *Active*
* 407.1 – If the patient record cannot be found in the database using the UID in the input

#### Image Service

A scan, or calibration, performed by a ClearView device produces a set of images. These images are the basis for all calculations and results produced. EPIC Central provides the image service to allow a device to upload images, where they are stored and can be viewed from anywhere in the context of a patient scan. The images are packaged as a compressed archive; there are multiple images in each archive and are referred to as an image set.

##### Method: Add

###### Purpose

This method allows a device to add a set of images to the EPIC Central database. *[Reqs: 4.6.1.9, 4.6.1.10]*

###### Function

This method accepts an image set from a device and stores it in the EPIC Central database. The device provides a UID for the image set, which is used to link the image set to the calibration or scan/treatment it is part of. The device also provides an MD5 hash (checksum) calculated over the contents of the image set before it is uploaded. The checksum is recalculated once the full image set is received and it must match the one calculated by the device.

Because image sets are very large files (multiple megabytes) they are not “posted” as part of the request. Instead they are streamed across the connection from the device to EPIC Central.

###### Inputs

This method requires three inputs: the UID for the image set, the MD5 checksum, and an input stream from which the image set is read.

###### Outputs

If this method completes successfully, it returns the long (64-bit) integer primary key for the entry in the database where the image set is stored.

###### Data

If this method completes successfully, it will add an entry to the ImageSet table in the database that will contain the binary image set as received from the device.

###### Errors

This method can produce the following errors returned to the client:

* 401.2 – If device is not authenticated
* 420 – If device’s state is not *Active*
* 430 – If no UID is provided for the image set
* 416 – If an image set already exists in the database for the specified UID
* 431 – If the checksum calculated on the received images set does not match the one provided by the device
* 502 – If an error occurs while reading the input stream containing the image set data

#### Calibration Service

A calibration is performed on a ClearView device to produce a baseline of environmental qualities that then become part of the analysis of a patient scan. A calibration is essentially a set of images taken with a controlled method. Because the calibration images are an essential part of any analysis, they need to be stored in the EPIC Central database where they are available for any further analysis or viewing of a scan/treatment.

The calibration service allows a device to add a calibration record to the EPIC Central database. The calibration record itself does not contain the images, simply a reference (by UID) to the image set for the calibration. Thus, the image set needs to be uploaded first, and then the calibration record that references it.

##### Method: Add

###### Purpose

This method allows a device to add a calibration to the EPIC Central database. *[Req: 4.6.1.9]*

###### Function

This method receives all information about a calibration from a device and creates an entry in the Calibration table in the EPIC Central database. The image set referenced by the calibration must already exist in the database. If the calibration already exists in the database, nothing is changed.

If a different scanner is attached to an existing ClearView system, protocol requires that the system reregister with EPIC Central. Reregistration will result in it becoming a different device, i.e. it will have a different device UID. When this occurs, there exists the possibility that a calibration performed with the previous scanner still needs to be uploaded. That calibration will reference a different device. This method makes special allowance for this case to allow the calibration to uploaded and added.

###### Inputs

This method requires a single input parameter, a *Calibration* object, containing the following fields:

* Calibration ID – not used as input
* GUID – the UID for the calibration; set by the device
* Timestamp – the date/time when the calibration was performed; must be a UTC value
* Performed By – the user name of the ClearView user that performed the calibration
* Image Set GUID – the UID of the image set that contains the images from the calibration
* Device GUID – the UID for the device that performed the calibration; this may not match the UID for the device that is performing the upload

###### Outputs

If this method completes successfully, it will return the same *Calibration* object it received as input, except the Calibration ID field will contain the primary key from the database for the record created.

###### Data

If this method completes successfully, it adds an entry for the calibration to the Calibration table in the database.

###### Errors

This method can produce the following errors returned to the client:

* 401.2 – If device is not authenticated
* 420 – If device’s state is not *Active*
* 406.2 – If the Device GUID is specified in the input and it cannot be found in the database
* 409 – If the image set reference cannot be found in the database
* 415 – If the calibration is already in the database but it is for a different device

#### Treatment Service

The end result when a patient is scanned using a ClearView device is a treatment record. This record includes (by reference) the set of images from the scan. It also includes all the analysis data produced by the analysis algorithms used to interpret the images.

The treatment service allows a device to add a treatment record along with its associated analysis data to the EPIC Central database. The treatment record itself does not contain the patient data, calibration data or image sets. These are all referenced (by UID). Thus, all these records must be uploaded (added) prior to adding the treatment.

##### Method: Add

###### Purpose

This method allows a device to add a treatment record to the EPIC Central database. *[Req: 4.6.1.10]*

###### Function

This method receives all information about a treatment from a device and creates an entry in the Treatment table in the EPIC Central database. The patient, calibration, and image sets referenced by the treatment must already exist in the database. If the treatment already exists in the database, an error will result.

In addition to the entry created in the Treatment table, the input contains a number of additional data set collections. Each entry in each of these collections is also added to the corresponding database table. All this data is opaque to this method; it simply pulls it from the input and puts it in the database.

###### Inputs

This method requires a single input parameter, a *Treatment* object, containing the following fields:

* Treatment ID – not used as input
* GUID – the UID of the treatment as assigned by the device
* Patient GUID – the UID of the patient the treatment is for
* Calibration GUID – the UID of the calibration used for the treatment
* Treatment Type – the type of the treatment
* Treatment Time - the date/time when the treatment was performed
* Performed By – the user name of the ClearView user who performed the scan
* Energized Image Set GUID – the UID of the image set containing the energized images
* Finger Image Set GUID – the UID of the image set containing the finger images; this is optional and will be null if there is none
* Software Version – the ClearView software version used
* Firmware Version – the scanner’s firmware version
* Analysis Time – the date/time when the analysis of the images was performed
* NB Analysis Results – some number of summary analysis results; these map directly to the NBAnalysisResult database table and are not detailed here
* Analysis Results – some number of raw analysis results; these map directly to the AnalysisResults database table and are not detailed here
* Severities – some number of severity calculation sets; these map directly to the Severity database table and are not detailed here
* Calculation Debugs – some number of calculation debug data sets; these map directly to the CalculationDebugData database table and are not detailed here
* Alcohol Question – boolean indicating whether or not the patient had consumed alcohol
* Wheat Question – boolean indicating whether or not the patient had consumed wheat
* Caffeine Question – boolean indicating whether or not the patient had consumed caffeine

The *Treatment* object is very large containing a number of different data sets and multiple occurrences of each of those. Very little of the data in the *Treatment* object is used directly by this method. It simply stores most of it in the database.

###### Outputs

If this method completes successfully, it will return the long (64-bit) integer primary key of the record added to the Treatment table in the database.

###### Data

If this method completes successfully, it will have added the following entry to the database:

* A single Treatment record which references a Patient record, a Calibration record and one or two ImageSet records
* Multiple NBAnalysisResult records, each referencing the Treatment record added
* Multiple AnalysisResult records, each referencing the Treatment record added
* Multiple Severity records, each referencing the Treatment record added
* Multiple CalculationDebugData records, each referencing the Treatment record added

###### Errors

This method can produce the following errors returned to the client:

* 401.2 – If device is not authenticated
* 420 – If device’s state is not *Active*
* 414 – If a treatment with the same UID already exists in the database
* 407.2 – If the patient referenced cannot be found in the database
* 408 – If the calibration referenced cannot be found in the database
* 409 – If either of the image sets referenced cannot be found in the database

### Messaging

A Service Administrator can create messages to send to ClearView devices. These message are then displayed to the user of the device. To get the messages, a device must make a request for currently pending messages. Each message is delivered only once to a device. *[Req: 4.6.7]*

The methods described here are all part of the *MessageService* class.

#### Method: GetCollection

##### Purpose

This method allows a device to get a collection of the messages pending for delivery to the device. *[Req: 4.6.1.11]*

##### Function

This method finds all messages pending for delivery to a device, creates an object to represent each and returns a collection of the messages. It marks each of the messages as delivered so they are not delivered again when the device again requests messages.

##### Inputs

This method has no input parameters. It knows the device because it must be authenticated.

##### Outputs

If this method completes successfully, it returns a *Messages* output which is a collection of zero or more *Message* objects. Each Message object has the following fields:

* Message ID – the primary key for the message in the EPIC Central database
* Type – the type of the message; it can be one of the following:
* Information – simple informational message
* Marketing – message for marketing purpose
* Sale – message announcing a sale on a product
* Attention – an alert that requires attention from the user
* Title – the title of the message
* Body – the body of the message
* Create Time – date/time when the message was created
* Start Time – date/time before which the message is not to be displayed; must be UTC value
* End Time – date/time after which the message is not to be displayed; must be UTC value

##### Algorithm

This method will find all entries in the *DeviceMessage* table that meet **all** the following criteria:

* Device ID of the entry matches the ID of the authenticated device
* Delivery time of the entry is null (it has not yet been delivered)
* The message is active (administrator has not removed it)
* The message start time matches the current time or is before the current time
* The message end time is either null (means there is no end time) or it matches the current time or is after the current time

Each message found will result in a *Message* object created and added to the output *Messages* collection.

##### Data

If this method completes successfully and returns any messages, the entry for each message in the *DeviceMessage* table will have its delivery time updated to the current time. Any entry with a non-null delivery time will not be delivered again.

##### Errors

This method can produce the following errors returned to the client:

* 401.2 – If device is not authenticated

### Exception Upload

ClearView devices report application exceptions to EPIC Central where they are stored to allow EPIC IT personnel to diagnose issues. These exceptions also update the operations map that allows a Service Administrator to watch activity at any or all locations.

#### Method: Create

##### Purpose

This method allows a device to report an exception to EPIC Central. *[Req: 4.6.1.12]*

##### Function

This method receives an exception from a device and stores the exception in the database.

##### Inputs

This method requires a single parameter, an *ExceptionLog* object that contains the following fields:

* Exception Log ID – primary key assigned by the database; not used as input
* Device ID – primary key for the device’s entry in the *Device* table; not used as input
* Remote Exception Log ID – ID of the exception on the client
* Title – summary title for the exception
* Message – message assigned when exception was created
* Stack Trace – dump of the stack trace when the exception occurred
* Log Time – date/time on the client when the exception occurred; must be UTC value
* User – name of the ClearView user on the client
* Form Name – name of the form on the client where exception occurred
* Machine Name – name of the client machine
* Machine OS – client machine’s operating system
* Application Version – the version of the application
* CLR Version – the version of the .NET Common Language Runtime on the client
* Memory Usage – snapshot of current memory usage when the exception occurred
* Received Time – date/time when the exception was received and logged by EPIC Central; must be a UTC value; not used as input

##### Outputs

If this method completes successfully, it will return the same *ExceptionLog* object it received as input except it will insert the Exception Log ID generated by the database as the primary key for the exception log record created and it will set the Received Time.

##### Data

If this method completes successfully, it will create a new entry in the *ExceptionLog* table to record the exception.

##### Errors

This method can produce the following errors returned to the client:

* 401.2 – If device is not authenticated

## Logging and System Diagnostics

Functionality has been put in place to record errors on multiple levels of operation. The lowest level of operation is where an error is caught at the core level, unhandled by any other code, and the server has no idea how to respond to that error. These types of errors are written to a text log. Higher level errors include exceptions uploaded from ClearView and errors that are ok to display to the user as feedback.

### Action Logger

The action logger executes at a low level, separate from other managed code. This logs every single page visit to EPIC Central, it shows when the page is loaded and when the execution of the page is completed. This makes all other events within the system easier to track. *[Req: 4.5.2.2]*

These are sample log entries created by the action logger:

2012-09-10 15:10:43,200 [502] DEBUG EPICCentral.Utilities.Filters.ActionLogAttribute [(null)] - Executed: GET - Menu - ListMenu - ::1 - 9/10/2012 3:10:42 PM

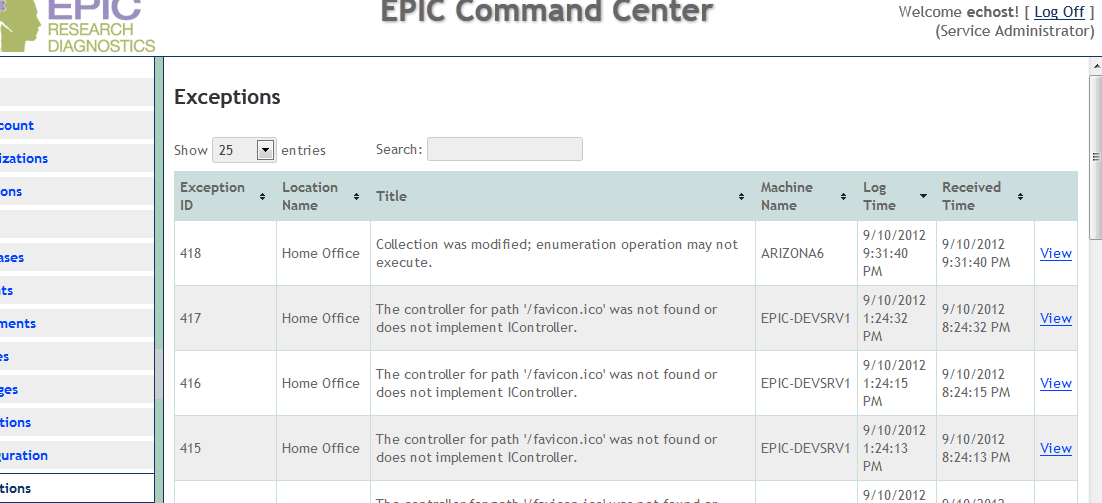
2012-09-10 15:10:43,278 [502] DEBUG EPICCentral.Utilities.Filters.ActionLogAttribute [(null)] - Executed: GET - Support - Messages - ::1 - 9/10/2012 3:10:42 PM

### Exceptions

Exceptions are uploaded from ClearView and viewable by a Service Administrator within EPIC Central. *[Req: 4.5.1.1]*

When an exception occurs, it is reflected on the *Operations* page.

An uploaded exception contains all of the required information and is searchable on the *Exceptions* page *[Req: 4.5.1.2]*:



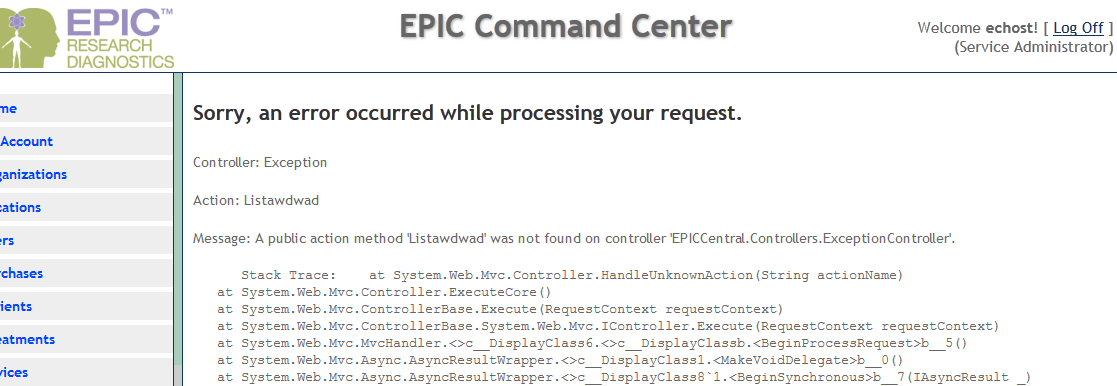
The *View* link can be used to view exception details such as the stack trace, memory usage and other exception details:

### Logging and Log4Net

Another logging mechanism is in place at the global level. Whenever an error occurs that is not handled directly by code a few steps occur *[Reqs: 4.5.2.1, 4.8.4.1, 4.8.4.2, 4.8.4.3]*:

1. It is logged to a file.
2. If it is an error, it is recorded in the same table as ClearView exceptions.
3. It is displayed to the user.

The first two steps are described above; the third is displayed in the image below:



Only administrators can see the exception details such as the stack trace.

For any user other than a Service Administrator, only the message is displayed under the “Sorry, …” heading.

### Auditing

The final level of logging is low level auditing. Every single change in the database is recorded to another table with the table name, a descriptive key, the old value, the new value, the date and time, and the user logged in when the change occurred.

## Device State

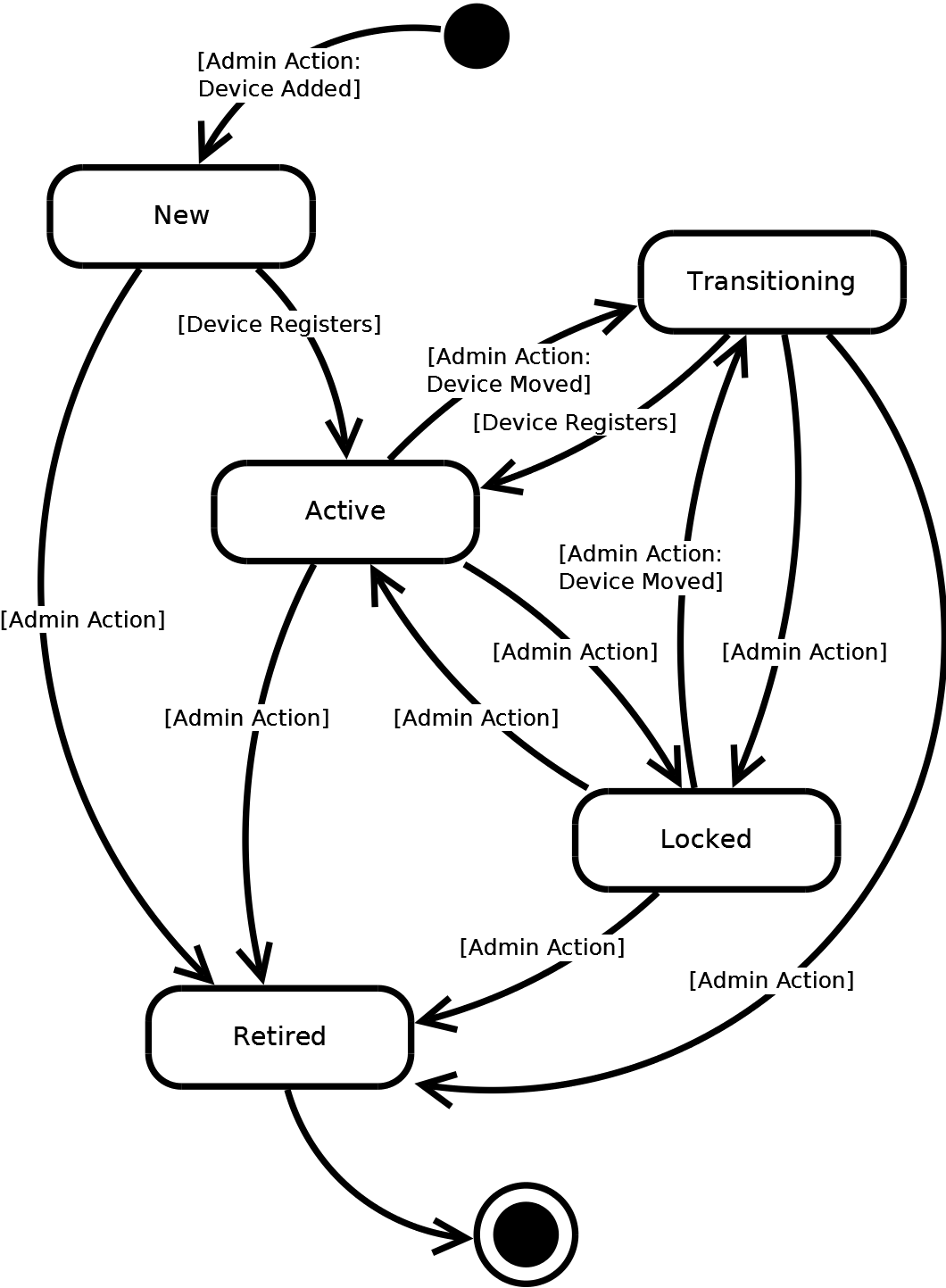
Device state isn’t specifically a component, but it is a key part of the lifecycle management of ClearView devices and needs to be covered in this specification.

### States and Transitions

A device defined in EPIC Central always has a defined state. Its state determines what actions it can perform and what other states it may move to. The following are the possible states for a device:

* *New* – The initial state set when the device is added to EPIC Central. No other state can be set when adding a device. After it has been added, its state can be changed.
* *Active* – The fully-operational state. In this state a device can perform all functions.
* *Transitioning* – This state means the device is temporarily out of service. There may be many reasons for using this state. The most common will be when a device is being moved from one physical location to another.
* *Locked* – This state means that the device is non-operational. It can still report status events but it cannot perform scans.
* *Retired* – This state means the device is permanently out of service. This is a final state. A device that is retired will never be used again.

The following state diagram shows the five possible device states and how a device may transition from one to another.



The only automatic state transitions are from either *New* or *Transitioning* to *Active* when a device registers. All others require administrative action. The following short sections expand on the diagram:

#### Into New State

When a device is added to EPIC Central, its initial state is always *New*. This state is set automatically; the user is not provided an option to set the initial state.

#### New to Active

When a new device finishes the registration process, its state is set to *Active* automatically.

#### New to Retired

There may be reasons to retire a device that is new even though it has never been put into service. One example would be a dead-on-arrival (DOA) device. If a device arrives at a customer location and will not run properly, it cannot be activated. If it is returned and fixed/updated, it will become a new/different device. So the original defined in EPIC Central will be retired.

#### Active to Transitioning

This state change requires administrative action. There may be many reasons to do this. A key point is that a device that is in the *Transitioning* state **must** reregister to become *Active* again. Reregistration will assign it a new unique identifier. Although, it’s the same device, it will be logically different from EPIC Central’s point of view. The most likely reason to use this state is to move a device from one location to another. When this occurs it needs to become a logically new device to maintain database integrity; the former logical device must remain attached to the original location.

#### Active to Locked

This state change also requires administrative action. We need the ability to prevent a device from doing scans. Reasons might be that the device has errors that prevent it from functioning correctly, or that there is a customer related issue, e.g. they owe EPIC money and have not paid. In this state, the device is still allowed to connect to EPIC Central and report status events. But it cannot perform a scan.

#### Transitioning to Active

When a transitioning device finishes the registration process, its state is set to *Active* automatically.

#### Transitioning to Locked

For the same reasons an *Active* device might be locked, so too can a *Transitioning* device. This requires administrative action.

#### Transitioning to Retired

A device that is being transitioned may need to be retired if something happens and it cannot become functional again. An example would be if a device is damaged while moving it from one physical location to another. In that case it would need to be returned to EPIC. If it is serviced, it will become a new device.

#### Locked to Active

If the reason for the device being locked is cleared, then an administrator can return it to *Active* state. For example, if a non-paying customer pays their bill, their device could become *Active* again.

#### Locked to Transitioning

If the reason for the device being locked is cleared, the administrator can unlock the device. But if the device was previously in the *Transitioning* state, it will return to the *Transitioning* state. It can’t go back to being *Active* immediately because a reregistration is required.

#### Locked to Retired

A locked device will be retired by an administrator when there is reason to do so. In the example of a customer not paying their bill, if they decide they do not want to be customer any longer, then their device would be retired and never be active again.

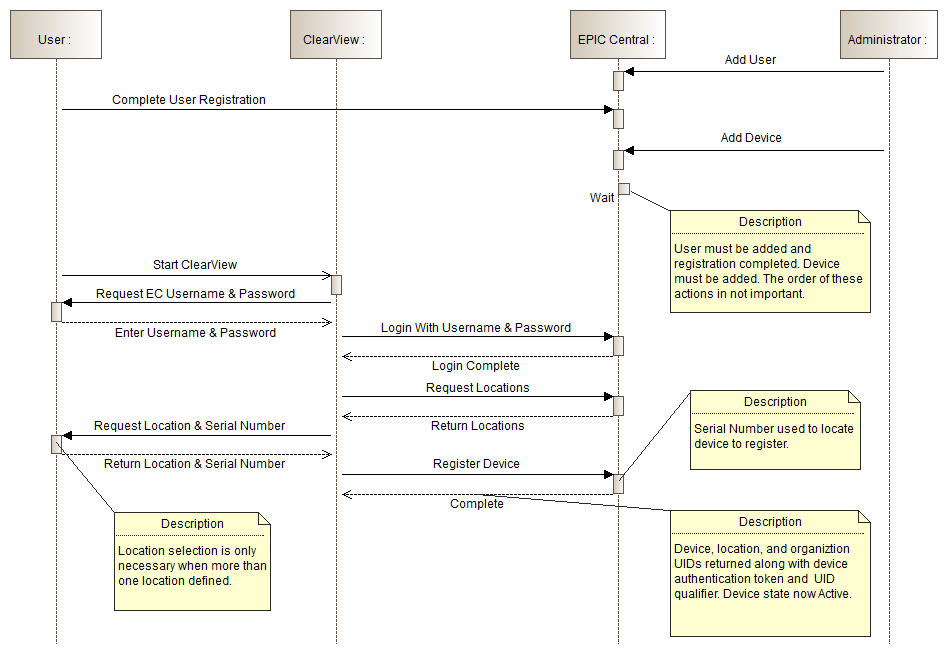
#### Retired is Final

Once a device enters the Retired state it is permanently out of service. The actual hardware may be refurbished and reenter service, but it will do so as a new and different device.

### Device Registration

Device registration is the process by which a ClearView device is connected to EPIC Central. Parts of this process have been discussed above. However, the entire process may not be clear. This section will discuss the full registration process.

The following sequence diagram shows the entire device registration process including prerequisites:



An EPIC Central user must perform the device registration using the ClearView application interface. This requires that the person doing the registration must have an EPIC Central user account. So prior to beginning the registration process, an administrator must add a user account for this person, and the user must then complete the user registration process to be able to log in to EPIC Central. Also, the Service Administrator must add the device to EPIC Central before it can be registered.

It doesn’t matter how long prior to device registration the user account is created and the device added. These are simply prerequisites that need to be done.

Assume a ClearView device will be shipped to a customer location. Someone will unpack the device and set it up. The ClearView computer will be started and the ClearView application will start. Since the device is new and has not been registered with EPIC Central, the application will begin the registration process by asking the user for his/her EPIC Central user name and password. The user will enter these and submit the entry form to the application.

The ClearView application will then establish a connection with the EPIC Central server and log in as the user and request the list of locations. EPIC Central will return the list of locations for the user’s organization. The application will then present this list of locations to the user and ask the user to select the location for the device and enter the device’s serial number from the device label itself. If there is only one location, the user may not be allowed to select a location.

When the user submits the form, the ClearView application will register the device with EPIC Central. When registration is complete, EPIC Central will return the device’s UID, the authentication token it will use to authenticate itself in the future, a UID Qualifier which is used to make all UIDs generated by the device unique across the EPIC Central ecosystem, and the UIDs for the location and organization the device is now registered with. The application will store these for future use. The device will be immediately in an *Active* state.

The above is a description of the process assuming there are no errors. The ClearView application must handle the following errors appropriately:

* Connection failure – a link across the Internet to EPIC Central may not be present.
* User login failure – the user name and password entered by the ClearView user may fail authentication.
* No locations found – there may be no locations configured for the user’s organization.
* No device found – the serial number entered does not match a device in EPIC Central.
* Location mismatch – the location the device was assigned to by the administrator when it was added does not match the location selected by the ClearView user.
* Invalid device state – the device in EPIC Central must be in the *New* or *Transitioning* state to allow registration; any other state will result in an error.

# Requirements Traceability Matrix

This section contains a matrix that cross references each requirement to the sections of this specification that describe how the requirement is met.

| Requirement | Requirement Summary | Section | Section Title |
| --- | --- | --- | --- |
| 4.1.1.1 | Organization types | 6.2.1 | Organization |
| 4.1.1.2 | Design for additional types | 6.2.1 | Organization |
| 4.1.1.3 | Only one Host organization allowed | 6.2.1 | Organization |
| 4.1.1.4 | Only SA can create an organization | 6.2.1 | Organization |
| 4.1.1.5 | SA can change organization name | 6.2.1 | Organization |
| 4.1.1.6 | OA can change name of his/her own organization | 6.2.1 | Organization |
| 4.1.2.1 | Location must have a name | 6.2.2 | Location |
| 4.1.2.2 | Location should have mailing address and telephone number | 6.2.2 | Location |
| 4.1.2.3 | SA can create/edit location for any organization | 6.2.2 | Location |
| 4.1.2.4 | OA can create/edit location for his/her own organization | 6.2.2 | Location |
| 4.1.2.5 | SA can edit geocode | 6.2.2 | Location |
| 4.1.2.6 | Automated lookup of geocode | 6.2.2 | Location |
| 4.1.3.1 | Device must be configured with EC | 6.2.5 | Device |
| 4.1.3.2 | Only SA can create/modify device | 6.2.5 | Device |
| 4.1.3.3 | Device must be assigned to one location | 6.2.5 | Device |
| 4.1.3.4 | Device must have serial number | 6.2.5 | Device |
| 4.1.3.5 | Device must have revision level | 6.2.5 | Device |
| 4.1.3.6 | SA may modify number of scans | 6.2.5 | Device |
| 4.1.3.7 | SA must provide notes when modifying number of scans | 6.2.5 | Device |
| 4.1.3.8 | Modifying number of scans creates record in purchase log | 6.2.5 | Device |
| 4.1.4.1 | Any person accessing EC must have a user account | 6.2.3 | User |
| 4.1.4.2.1 | User account must be created by an administrator | 6.2.3 | User |
| 4.1.4.2.2 | SA able to configure user for any organization | 6.2.3 | User |
| 4.1.4.2.3 | OA able to configure user for his/her own organization | 6.2.3 | User |
| 4.1.4.2.4 | Required data when creating user account | 6.2.3 | User |
| 4.1.4.2.5 | Validation of required data when creating user account | 6.2.3 | User |
| 4.1.4.2.6 | Email format validation | 6.2.3 | User |
| 4.1.4.2.7 | Newly created user account not usable until registration is completed | 6.2.3 | User |
| 4.1.4.2.8 | Notify user of account creation | 6.2.3 | User |
| 4.1.4.2.9 | Notification must uniquely identify account | 6.2.3 | User |
| 4.1.4.3.1 | User must complete registration | 6.2.4.2.3 | User Registration |
| 4.1.4.3.2 | User must uniquely identify account | 6.2.4.2.3 | User Registration |
| 4.1.4.3.3 | User must choose valid user name | 6.2.4.2.3 | User Registration |
| 4.1.4.3.4 | User must create valid password | 6.2.4.2.3 | User Registration |
| 4.1.4.3.5 | User must confirm password | 6.2.4.2.3 | User Registration |
| 4.1.4.3.6 | EC must validate user name format and uniqueness | 6.2.4.2.3 | User Registration |
| 4.1.4.3.7 | EC must validate password | 6.2.4.2.3 | User Registration |
| 4.1.4.3.8 | EC must validate confirmed password matches | 6.2.4.2.3 | User Registration |
| 4.1.4.3.9 | If invalid, EC must inform user and allow corrections | 6.2.4.2.3 | User Registration |
| 4.1.4.3.10 | When valid, EC must create account and log user in | 6.2.4.2.3 | User Registration |
| 4.1.5.1 | Access to functions will be role based | 6.1.1.2.1 6.1.1.2.2 | Menu Allow/Deny |
| 4.1.5.2 | User must have exactly one role | 5.7 6.2.3 | Roles User |
| 4.1.5.3.1 | Minimum of three named roles defined | 5.7 | Roles |
| 4.1.5.3.2 | SA role not restricted to an organization | 5.7 | Roles |
| 4.1.5.3.3 | OA role restricted to the user’s organization | 5.7 | Roles |
| 4.1.5.3.4 | Simple User role restricted to user’s own settings | 5.7 | Roles |
| 4.1.5.4.1 | Role assigned by administrator | 5.7 6.2.3 | Roles User |
| 4.1.5.4.2 | SA can assign any role to any user | 5.7 6.2.3 | Roles User |
| 4.1.5.4.3 | OA can assign OA or Simple User role to any user in his/her organization | 5.7 6.2.3 | Roles User |
| 4.1.5.4.4 | Simple User cannot assign roles | 5.7 6.2.3 | Roles User |
| 4.1.5.4.5 | Role assigned when user created | 5.7 6.2.3 | Roles User |
| 4.1.5.4.6 | Administrator can change role anytime | 5.7 6.2.3 | Roles User |
| 4.1.5.4.7 | Role change takes effect on next user action | 5.7 6.2.3 | Roles User |
| 4.1.6.1 | SA able to disable/re-enable device access anytime | 6.2.7 | System Settings |
| 4.1.7.1 | SA able to create purchase rate structure | 6.2.7 | System Settings |
| 4.1.7.2 | Rate structure has multiple price points | 6.2.7 | System Settings |
| 4.1.7.3 | Price point has effective date | 6.2.7 | System Settings |
| 4.2.1.1 | All users must authenticate | 6.2.4.2.1 | Log On |
| 4.2.1.2 | No persistence of credentials by browser | 6.2.4.2.1 | Log On |
| 4.2.1.3 | Log-off function to allow user log off | 6.2.4.2.2 | Log Off |
| 4.2.1.4 | Session cleared on log off | 6.2.4.2.2 | Log Off |
| 4.2.1.5 | 60-minute session timeout | 6.2.4.2.2 | Log Off |
| 4.2.2.1 | Operations restricted to SA role | 6.4 | ClearView Earth |
| 4.2.2.2 | Daily counts to be displayed | 6.4 | ClearView Earth |
| 4.2.2.3 | Aggregate counts to be displayed | 6.4 | ClearView Earth |
| 4.2.2.4 | Map with device locations shown | 6.4 | ClearView Earth |
| 4.2.2.5 | Icons show most recent activity | 6.4 | ClearView Earth |
| 4.2.2.6 | User can pan map | 6.4 | ClearView Earth |
| 4.2.2.7 | User can zoom in and out | 6.4 | ClearView Earth |
| 4.2.2.8 | Device pop-up when user click icon | 6.4 | ClearView Earth |
| 4.2.3.1 | User able to view patient records within organization | 6.3.2.1 | Patient Search |
| 4.2.3.2 | Minimum patient search requirements | 6.3.2.1 | Patient Search |
| 4.2.3.3 | SA able to view/search any patient | 6.3.2.1 | Patient Search |
| 4.2.3.4 | User able to view treatment for any patient user has access to | 6.3.2.4 | Treatment Search |
| 4.2.4.1 | User able to view treatment records within organization | 6.3.2.4 | Treatment Search |
| 4.2.4.2 | Minimum treatment search requirements | 6.3.2.4 | Treatment Search |
| 4.2.4.3 | SA able to view/search any treatment | 6.3.2.4 | Treatment Search |
| 4.2.4.4 | Treatment view modeled after way device shows treatment | 6.3.2.4 | Treatment Search |
| 4.2.5.1.1 | Third-party payment gateway | 6.5.3 | Managing Credit Cards |
| 4.2.5.1.2 | EC doesn’t store usable credit card information | 6.5.3 | Managing Credit Cards |
| 4.2.5.1.3 | EC may store subset of credit card information | 6.5.3 | Managing Credit Cards |
| 4.2.5.1.4 | EC may store and present name on card and part of billing address | 6.5.3 | Managing Credit Cards |
| 4.2.5.1.5 | CCV is required on first card usage | 6.5.3 | Managing Credit Cards |
| 4.2.5.1.6 | Subsequent card usage possible with stored information | 6.5.3 | Managing Credit Cards |
| 4.2.5.2.1 | User able to purchase for any device in assigned locations | 6.5.2 | Making a New Purchase |
| 4.2.5.2.2 | OA able to purchase for any device in any of organization’s locations | 6.5.2 | Making a New Purchase |
| 4.2.5.2.3 | Scans available increase when purchase completed | 6.5.2 | Making a New Purchase |
| 4.2.5.2.4 | Scans transferrable to another device | 6.5.4 | Transferring Scans |
| 4.2.5.2.5 | Log of purchase and transfers kept | 6.5.1 | Purchase History |
| 4.2.5.2.6 | Minimum log search requirements | 6.5.1 | Purchase History |
| 4.2.6.1.1 | User may change password | 6.2.4.2.4 | Change Password |
| 4.2.6.1.2 | Required entries when changing password | 6.2.4.2.4 | Change Password |
| 4.2.6.1.3 | Required validations when changing password | 6.2.4.2.4 | Change Password |
| 4.2.6.1.4 | Password takes effect on next login | 6.2.4.2.4 | Change Password |
| 4.2.6.2.1 | User able to reset forgotten password | 6.2.4.2.5 | Password Reset |
| 4.2.6.2.2 | User must provide email to reset | 6.2.4.2.5.1 | Starting the Reset Process |
| 4.2.6.2.3 | User may provide user name to reset | 6.2.4.2.5.1 | Starting the Reset Process |
| 4.2.6.2.4 | Email sent to email address on account | 6.2.4.2.5.1 | Starting the Reset Process |
| 4.2.6.2.5 | Account restricted until reset complete | 6.2.4.2.5.1 | Starting the Reset Process |
| 4.2.6.2.6 | All accounts restricted if user only provides email address and more than one account | 6.2.4.2.5.3 | Starting the Reset Process, Multiple Accounts |
| 4.2.6.2.7 | User must complete reset on EC using information in email received | 6.2.4.2.5.2 | Completing the Reset Process |
| 4.2.6.2.8 | EC allows user to enter and confirm new password | 6.2.4.2.5.2 | Completing the Reset Process |
| 4.2.6.2.9 | New password takes effect immediately | 6.2.4.2.5.2 | Completing the Reset Process |
| 4.3.1.1 | Device must register to operate | 6.7.2 | Registration |
| 4.3.1.2 | Registration required for new and moved devices | 6.7.2 | Registration |
| 4.3.1.3 | User performing registration must have EC user account | 6.7.2 | Registration |
| 4.3.1.4 | Device must be assigned to exactly one location | 6.7.2 | Registration |
| 4.3.1.5 | Assigned location must be location in user’s organization | 6.7.2 | Registration |
| 4.3.1.6 | User must provide device serial number | 6.7.2 | Registration |
| 4.3.1.7 | Device fully operational after registration complete | 6.7.2 | Registration |
| 4.3.2.1 | SA able to add new device | 6.2.5.2.3 | Add a New Device |
| 4.3.2.2 | SA able to set state to allow re-registration | 6.2.5.2.4 | Edit an Existing Device |
| 4.3.2.3 | SA able to lock device to prevent scans | 6.2.5.2.4 | Edit an Existing Device |
| 4.3.2.4 | SA able to unlock device | 6.2.5.2.4 | Edit an Existing Device |
| 4.3.2.5 | SA able to permanently take device out of service | 6.2.5.2.4 | Edit an Existing Device |
| 4.3.2.6 | Device to be re-registered cannot scan | 6.7.4.1 | Method: Validate |
| 4.3.2.7 | Locked device cannot scan | 6.7.4.1 | Method: Validate |
| 4.3.2.8 | Device taken out of service cannot authenticate | 6.7.4.1 | Method: Validate |
| 4.4.1.1 | Any authenticated user can request support | 6.6 | Support Center |
| 4.4.1.2 | Newly create support request sent to designated email address | 6.6 | Support Center |
| 4.4.1.3 | Minimum support request data | 6.6 | Support Center |
| 4.4.1.4 | Message recipient able to respond with another message | 6.6 | Support Center |
| 4.4.1.5 | User can view their requests and responses to those requests | 6.6 | Support Center |
| 4.4.1.6 | User can remove messages | 6.6 | Support Center |
| 4.4.1.7 | New requests cause operations view to update | 6.6 | Support Center |
| 4.4.2.1 | Message can be created and sent to any device, location or organization | 6.2.6 | Message |
| 4.4.2.2 | Message sent to location is sent to every device at that location | 6.2.6 | Message |
| 4.4.2.3 | Message sent to organization is sent to every device in that organization | 6.2.6 | Message |
| 4.4.2.4 | Message delivered only once | 6.2.6 | Message |
| 4.4.2.5 | Message can be changed before delivery | 6.2.6 | Message |
| 4.4.2.6 | Message can be removed before delivery | 6.2.6 | Message |
| 4.4.2.7 | Only SA can create, change or remove messages | 6.2.6 | Message |
| 4.5.1.1 | Only SA can view uploaded exceptions | 6.8 | Logging and System Diagnostics |
| 4.5.1.2 | Minimal exception details | 6.8.2 | Exceptions |
| 4.5.1.3 | Minimum exception search requirements | 6.8.2 | Exceptions |
| 4.5.2.1 | Application exceptions logged | 6.8.3 | Logging and Log4Net |
| 4.5.2.2 | Actions and events logged | 6.8.1 | Action Logger |
| 4.6.1.1 | WS for registering new device | 6.7.2.2 | Method: RegisterDevice |
| 4.6.1.2 | WS for re-registering device | 6.7.2.2 | Method: RegisterDevice |
| 4.6.1.3 | WS for resetting device credentials | 6.7.2.3 | Method: ResetAuthenticationKey |
| 4.6.1.4 | WS for validating ability to scan | 6.7.4.1 | Method: Validate |
| 4.6.1.5 | WS for getting number of scan available | 6.7.4.2 | Method: GetCount |
| 4.6.1.6 | WS for reporting scan completion | 6.7.4.3 | Method: ScanComplete |
| 4.6.1.7 | WS for adding a patient | 6.7.5.1.1 | Method: Add |
| 4.6.1.8 | WS for updating a patient | 6.7.5.1.2 | Method: Update |
| 4.6.1.9 | WS for adding a calibration record | 6.7.5.3 6.7.5.2 | Calibration Service Image Service |
| 4.6.1.10 | WS for adding a treatment record | 6.7.5.4 6.7.5.2 | Treatment Service Image Service |
| 4.6.1.11 | WS for requesting messages | 6.7.6 | Messaging |
| 4.6.1.12 | WS for adding exception record | 6.7.7 | Exception Upload |
| 4.6.1.13 | WS for reporting status | 6.7.3 | Events |
| 4.6.2 | Every invocation is authenticated | 6.7.1 | Authentication |
| 4.6.3 | Authentication is done with user name and password during registration | 6.7.2 | Registration |
| 4.6.4 | After registration, device authenticates | 6.7.1 | Authentication |
| 4.6.5 | Device authentication highly secure | 6.7.1 | Authentication |
| 4.6.6 | EC tracks number of scan available | 6.7.4.3 | Method: ScanComplete |
| 4.6.7 | Message is delivered only once | 6.7.6 | Messaging |
| 4.6.8 | Atomic transactions required when changing data | 5.2 | Data Layer |
| 4.7.1.1 | User name length | 6.2.4.2.3 | User Registration |
| 4.7.1.2 | No space allowed in user name | 6.2.4.2.3 | User Registration |
| 4.7.1.3 | Password length | 6.2.4.2.3 | User Registration |
| 4.7.1.4 | No space allowed in password | 6.2.4.2.3 | User Registration |
| 4.7.1.5 | Password composition | 6.2.4.2.3 | User Registration |
| 4.7.2.1 | Unique IDs required | 6.2.1 6.2.2 6.2.5 | Organization Location Device |
| 4.7.2.2 | EC guarantees device-assigned IDs are unqiue | 6.2.5 6.7.2.2 | Device Method: RegisterDevice |
| 4.7.3.1 | Auditable history data not deleted | 6.2.1.3 6.2.2.3 6.2.3.3 6.2.5.3 | Organization, Data Location, Data User, Data Device, Data |
| 4.7.3.2 | Data no longer needed marked inactive | 6.2.1 6.2.2 | Organization Location |
| 4.8.1.1 | Public Internet access only via HTTPS | 5.6 | Access Restrictions |
| 4.8.1.2 | System console access only via local network | 5.6 | Access Restrictions |
| 4.8.1.3 | System console access only by Windows administrators | 5.6 | Access Restrictions |
| 4.8.2.1 | IE 9 and above supported | 5.5 | User Interface |
| 4.8.2.2 | Firefox 11 and above supported | 5.5 | User Interface |
| 4.8.2.3 | Chrome 18 and above supported | 5.5 | User Interface |
| 4.8.2.4 | Safari 5.1 and above supported | 5.5 | User Interface |
| 4.8.3.1 | Standard internationalization | 5.8 6.2.3.2.5 | Internationalization & Localization Edit One’s Own Settings |
| 4.8.3.2 | Localization for US-English | 5.8 | Internationalization & Localization |
| 4.8.3.3 | User selects localization based on location or user setting | 5.8 6.2.3.2.5 | Internationalization & Localization Edit One’s Own Settings |
| 4.8.3.4 | US-English is default localization | 5.8 6.2.3.2.5 | Internationalization & Localization Edit One’s Own Settings |
| 4.8.4.1 | User errors detected early | 6.8 | Logging and System Diagnostics |
| 4.8.4.2 | Error message clear and consistent | 6.8.3 | Logging and Log4Net |
| 4.8.4.3 | Error messages include hints | 6.8.3 | Logging and Log4Net |
| 4.8.5.1 | Web UI style consistent with EPIC corporate website | 5.5 | User Interface |
| 4.8.5.2 | Link to EPIC corporate website | 6.1.5 | Menu |
| 4.8.5.3 | Consistent presentation of UI facets | 5.5 6.1.3 | User Interface DataTables |
| 4.8.5.4 | Functions unavailable to user not presented to user | 6.1.1.2.1 6.1.1.2.2 | Menu Allow/Deny |