Appendix C
Systems Reviewed in Detail

ClearHealth

Project Background: Of the FOSS EHR systems reviewed in detail, ClearHealth is the newest. David Uhlman developed the ClearHealth EHR in 2003 when he served as the CEO of a business software firm named Uversa Consulting Inc. Since then, ClearHealth Inc. has been spun off from Uversa, and David Uhlman has assumed the CEO position of the new corporate entity, which continues to develop the ClearHealth EHR.

Some members of the ClearHealth development team previously worked on the FreeMed and OpenEMR systems, and the ClearHealth team has been able to leverage this experience to build a new application from the ground up. Development first focused on practice management features, especially scheduling, patient demographics, document management, and billing. ClearHealth 1.x was primarily a practice management system with a limited amount of clinical encounter support. The latest release, ClearHealth 2.0, has significantly expanded the clinical functionality of the system.

Functionality: ClearHealth has provided effective practice management functionality since its v1.x releases. The calendar and scheduling features are intuitive and easy to use compared to the other FOSS EHRs that this study reviewed. The product has strong billing support. In addition, ClearHealth offers an integrated reporting system, a feature that is not offered by the other FOSS EHRs this study saw.

ClearHealth v2.0, released in the summer of 2007, adds some support for clinical data management. Relative to its mature practice management capabilities, however, ClearHealth's EMR features are new and as-yet untested in the field. One of the product's most significant gaps is the lack of native support for standardized terminologies, such as LOINC and SNOMED. Given its relatively short history, it is reasonable to expect refinement and enhancements to the EHR functionalities in the coming months. Table 1 and Table 2 list the set of features available in the ClearHealth v2.0 release at this time. A screenshot of the application appears in Figure 1.

Table 1 – ClearHealth Functionality (1 of 2)

Area	Feature	Support	Support Details
Practice Management	Scheduling/ Calendar	■	A fully featured interactive calendar that supports all patient appointment scheduling needs (this is based on the Ajax technology).
	Patient Registration	■	Full patient demographics entry is supported. An unlimited number of demographic elements may be captured for each patient.
	Billing System	■	The application offers a fully integrated billing system that supports online eligibility verification for Medicare/Medicaid recipients. It allows direct-to-payer billing as well as electronic billing via a billing service. Sliding scale billing for Federally Qualified Health Centers (FQHCs) is also supported.
	Reporting	■	A fully configurable reporting engine is included as part of the core application. Base installation has a set of administrative reports (approx 40). Some encounter-based reports are available, but these are of limited complexity. Reports can be added and edited using the reporting engine but significant knowledge of HTML, CSS, SQL, and the underlying data model is required to do so. Reports may be run on-demand or as regularly scheduled events.
Electronic Medical Record	Problem Lists	■	A list of active problems is available for each patient in the "Criticals Palette" section of the patient dashboard. The "Criticals Palette" can be configured to show other quick lists such as self management goals, physical exams, allergies, and medications. Problem list contents are populated from a configurable list of common problems, but are not encoded in ICD-9.
	Note Generation	■	The application supports SOAP-style clinical notes. Fields may be populated with free-text or via fill-in templates. Fill-in templates can be edited, but some development expertise is required to do so. ICD-9 and CPT-4 coding may be performed within the context of the "Plan" form of the SOAP note for billing documentation.
	e-Prescribing	▒	E-prescribing is not a core feature of the application. However, interfaces with SureScripts and with Allscripts' TouchScript® services have been developed and are available (although not yet certified by the e-prescribing networks).
	Standard Prescribing	▒	Prescriptions may be recorded in the application and printed to a local printer or faxed via the application's faxing gateway. Entering prescription information is currently free-text based with no standardized vocabulary, order entry support, or drug interaction checking. The application provides a medication list to assist providers with selecting a drug name.
	Electronic Lab Orders		ClearHealth does not yet have a standard electronic lab ordering system. An interface to Quest Diagnostics is being developed for release in the third quarter of 2007.
	Standard Lab Orders	■	Lab orders may be recorded and printed with printing templates that are specific to various labs.
	Electronic Lab Reporting	■	The application supports an electronic real-time interface with Quest Diagnostics and a few regional hospitals for the receipt of electronic laboratory data in HL7 format.
	Document Management	■	A document storage system is included in the core application. It can be connected to a variety of source systems including copy machines, faxes, etc. Documents may also be uploaded from the local file system via the browser. Documents can be "filed" into a customizable set of document categories. Practice sites using the ClearHealth Online hosted service are limited to the amount of space available for document storage.

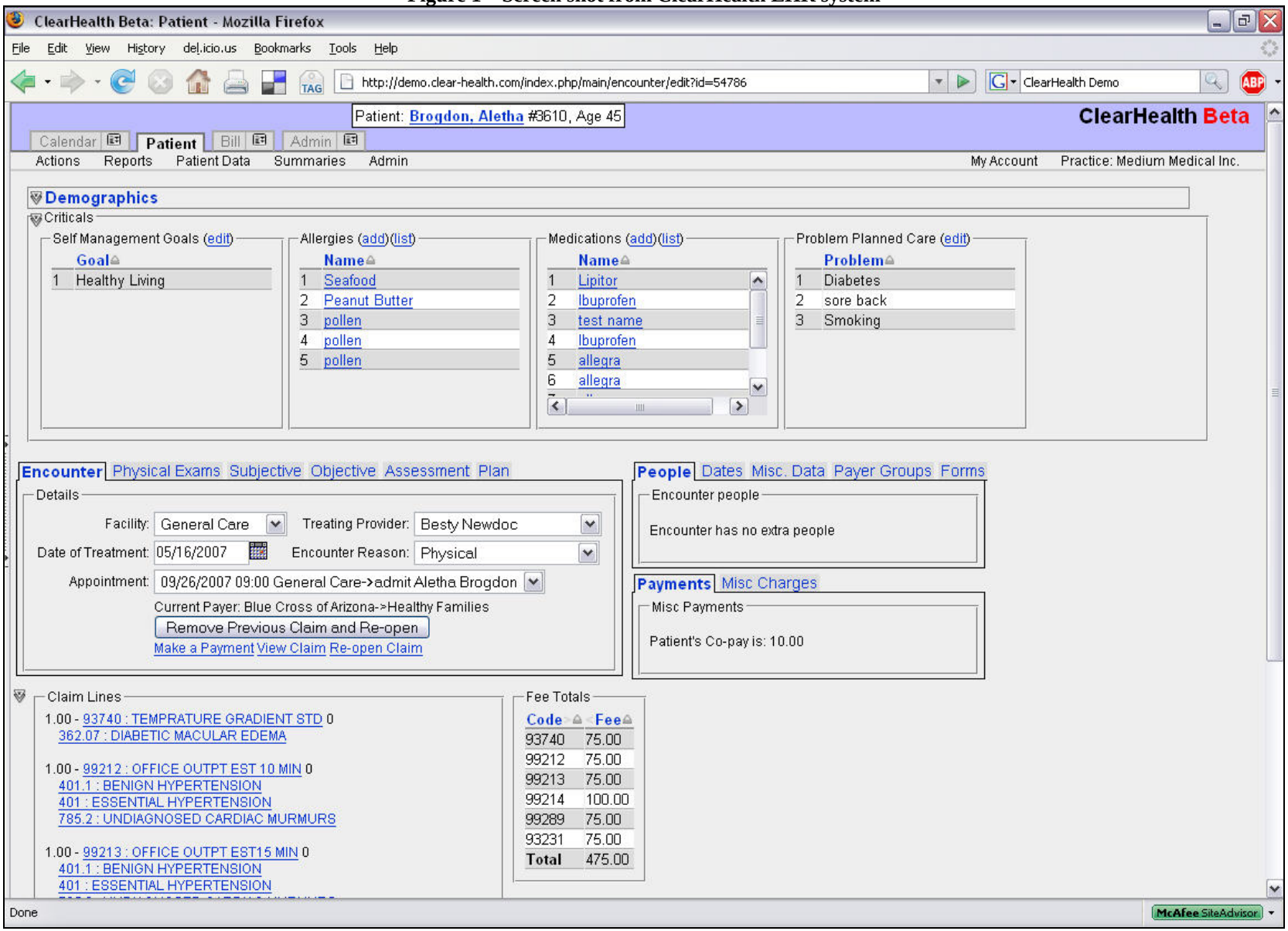
Legend: ■ Functionality Supported ▒ Some Functionality Supported □ Not Supported

Table 2 – ClearHealth Functionality (2 of 2)

Area	Feature	Support	Support Details
Electronic Medical Record [cont.]	Decision Support		Default decision support is not included in the application. However, ClearHealth supports integration with 3rd party proprietary decision support modules to meet specific health decision support needs, if this functionality is desired by a practice. However, these solutions are not compatible with GPL, are not available for download from the public code repository, and require the payment of significant licensing fees for their use.
	Disease Management		There is disease management support in the most recent version of the application. The ClearHealth reporting engine may be used to generate reports similar to those of the CVDems and PECS disease management systems.
General	Interoperability/ API		The application can send and receive data in the CCR data format. The application also provides a basic HL7 interface for the receipt of HL7 data. The Mirth open source HL7 interface engine may be integrated with the application for complex HL7 interoperability. The application also offers a web services wrapper that can be used in wide variety of ways. It supports SOAP, XMLRPC, and REST formats for accessing the web services layer.
	Standardized Vocabularies		ICD-9 and HCPCS codes are supported with the core installation. CPT4 codes are supported and can be installed into the ClearHealth reference database after purchasing a separate license. No clinical vocabularies (e.g., LOINC, SNOMED, and RxNorm) are natively supported by the application but LOINC and SNOMED may be added to the application's reference database during installation if required by the user. The application supports HL7 and the CCR clinical document format.
	Other		• Allergy and Immunization documentation is supported (free-text). • Allows practice to document patient disease education • Ability to track inbound and outbound referrals • Can connect with wireless devices (i.e., vitals carts) for collection of "Objective" encounter data

Legend: Functionality Supported; Some Functionality Supported; Not Supported

Figure 1 – Screen shot from ClearHealth EHR system



Installation Options and Architecture: ClearHealth offers three installation options: ClearHealth Office Online; ClearHealth Office Onsite; and ClearHealth Advantage.

ClearHealth office Online is the hosted option. It is aimed at the small practice (fewer than ten physicians or 20,000 encounters per year), and entails a user-subscription model. ClearHealth Office Onsite is the mid-level offering. It can be hosted at the practice site or at a data center. It is more scalable than the ClearHealth Online product and is shipped on a pre-configured hardware appliance. It is intended for sites with around 10 physicians or less than 45,000 encounters per year. ClearHealth Advantage is the enterprise version of the system, aimed at larger, multi-site practices. This option is intended for customers with significant customization or integration requirements. ClearHealth works with the practice site(s) to fully integrate this solution into their existing network infrastructures and clinical workflows.

ClearHealth is built on the LAMP stack. It was written from scratch for PHP5 (the most recent version), which provides enhanced security and data-handling support.

Support Services: The main source of revenue for ClearHealth comes from the support services that it provides to its commercial customers. ClearHealth offers a full set of such services on a contracted basis, including installation, configuration, customization, maintenance, and support. ClearHealth offers three levels of support contracts (Incident, Gold, and Platinum). The most common customization service offered by ClearHealth is the conversion of existing paper forms into web-based forms. These forms can be used in the context of an encounter to collect information of special interest to the practice or information that meets a requirement of a certain medical specialty.

Open Source License: Most code written by ClearHealth is licensed under GPL, although certain libraries used by the application are under different licenses, such as LGPL and MPL. The source code for certain modules is not made available in the public code repository due to special third-party licensing provisions or NDAs. This source code is maintained in a separate private repository that is not accessible to the public. The code is released to only those customers who purchase the appropriate third-party license or are covered by the necessary NDAs.

Development Community: Most of the programming is done by ClearHealth staff, consisting of roughly ten full-time developers. ClearHealth also has a fairly large open source development community (there are approximately 280 active forum members). The project utilizes a user forum to capture community feedback and feature requests. It incorporates these requirements into its development roadmap as best serves its business needs and commercial engagements. Source code submissions from outside developers are accepted, but these submissions are reviewed before they are incorporated into the public code repository. In any case, such contributions represent a small fraction of the current code base (for example, of the 12,000 lines of code submitted by the open source development community last year, approximately half were ultimately accepted into the public repository).

Related Projects: MirrorMed is a fork of the ClearHealth v1.5 code base that includes only the practice management functionality. This code base is maintained and updated by SynSeer, a company run by Fred Trotter, who was an early contributor and project manager for the ClearHealth project at Uversa. SynSeer offers service and support to customers who implement

the MirrorMed practice management application. One notable feature of the MirrorMed application is an interface to the Medical Manager Practice Management system.

ClearHealth and SynSeer engage in very little interaction and coordination at the present time. SynSeer makes enhancements to the MirrorMed code base as needed for its own customer base. ClearHealth is moving forward with its v2.0 release independently of SynSeer. Although ClearHealth does receive occasional coding contributions from Fred Trotter, it does not otherwise review and integrate code produced for MirrorMed.

Analysis: ClearHealth is the largest commercial FOSS EHR vendor on the market. The company has about 20 employees, half of whom are developers. They have the resources and financial backing to quickly add features and to execute significant implementations of their EHR. The focus of their business is on larger, multisite medical practices, and they hope to enter the hospital/inpatient market. Their hosted service may be more affordable for small practices, but the limited customization available with this option and the reluctance of some physicians to remotely store their patient data may limit adoption.

ClearHealth has a strong practice management system, which is mature and offers a full range of capabilities. The EMR capabilities are newer and less mature, and there will likely be refinements and expansions to this part of the application over the next year or two. As with most of the FOSS EHR products this study reviewed, there is no limited decision support built into the application. In addition, there is limited support for clinical coding standards, so that the later addition of decision-support features may require significant enhancements.

Final Word: ClearHealth is a good solution for medium to large practices and community health clinics that have funding available to sponsor the development of missing EHR features (although the costs of such enhancements are not excessive).

FreeMed

Project Background: FreeMed is the oldest of the FOSS EHR projects. Work on the project under its current architecture started in 1998, although it was based on previous work of Irving J. Buchbinder, D.P.M., DAPBS, that dates back to the mid-1980's. Dr. Buchbinder and his son, Jeff Buchbinder, are the two project leads for the FreeMed EHR. Jeff Buchbinder is the chief architect and lead developer of the FreeMed code-base. In addition to the contributions of the Buchbinders, Volker Bradley, M.D., Daniel D'Andrea, M.D., Frank Vailer, Ph.D., and Philipp Meng have contributed their clinical and technical expertise to the development of this application.

The FreeMed project is organized under a nonprofit model. The FreeMed Foundation, a 501(c)(3) not-for-profit corporation, manages the project. The foundation consists of a board of directors, which guides the development road-map of FreeMed, and a Community Clinical Advisory Board, which provides clinic expert knowledge for development projects.

Functionality: FreeMed has a good combination of practice management and EHR functionalities. Of note is the project's strong support for data standards compared to that of the other FOSS EHR projects this study reviewed. For example, FreeMed provides native support for ICD-9, LOINC, HCPCS, HL7, CCR, as well as the ability to load CPT4 and SNOMED codes once the necessary licenses have been obtained. FreeMed has good support for laboratory

interfacing (using HL7 and the Mirth open-source interface engine). It has made some progress in the area of disease management with a system that allows entries in problem lists to be semantically tagged (similar to how pictures are tagged using the Flickr photo sharing web site), which allow classification of diagnoses into desired categories (e.g., pulmonary condition). Any number of tags can be applied to each diagnosis, and these tags can then be used for reporting purposes. FreeMed also offers a solid interoperability solution with their API (referred to as its data relay system), which is available for a number of common programming languages.

FreeMed lacks general decision support and electronic ordering capabilities, a deficiency of all the FOSS EHR systems this study reviewed. The application does not offer an integrated reporting engine and reporting is dependent on the use of third-party reporting tools. FreeMed is in the process of moving from the current version 0.8.4 to the next major version 0.9. This major version change will reportedly include a significant number of enhancements, including a reworking of the architecture to separate the display logic from the underlying data model, as well as an update to the user interface and look and feel.

The features listed in Table 3 and Table 4 are that of the current v0.8.4 release. A screenshot of the application appears in Figure 2.

Table 3 – FreeMed Functionality (1 of 2)

Area	Feature	Support	Support Details
Practice Management	Scheduling/ Calendar	■	The application provides a scheduling calendar and a group scheduler.
	Patient Registration	■	Standard patient demographics, contact information, insurance coverage, preferred provider, preferred pharmacy, and personal notes may be captured for each patient
	Billing System	■	Billing functionality is handled by ReMITT, a medical billing module. It has a claims manager that handles accounts receivable and can post checks. ReMITT can route submission of claims through a clearing house. A minor feature allows for the definition of required and disallowed procedure codes for claim justification assistance. FreeMed supports sliding scale billing for Federally Qualified Health Centers (FQHCs).
	Reporting	▒	A small set of administrative reports are available. The display of report data may be graphical or textual. The application does not provide report-generation tools at the end-user level.
Electronic Medical Record	Problem Lists	■	Chronic and current problem lists are displayed for patients.
	Note Generation	■	SOAP note encounter information may be recorded. Free-text and templates can be used for note population. There are plans to utilize existing diagnoses to drive the template pick-list, but this is not currently supported. The FreeMed foundation recognizes the need to be able to normalize data.
	e-Prescribing		Currently the application does not support e-Prescribing but is in the process of working with e-Prescribing service to negotiate the details around submitting prescriptions in the NCPDP format.
	Standard Prescribing	▒	The application can currently print, fax, and email prescriptions to a pharmacy of the patient's choosing. Prescription order entry is assisted by an online formulary containing drug names, strengths, and forms.
	Electronic Lab Orders		The application provides no native support for electronic laboratory orders. However, if a FreeMed implementation has been integrated with the Mirth interface engine, laboratory orders may be sent electronically via an available Mirth connector.
	Standard Lab Orders	■	Laboratory orders can be recorded and, with an appropriate printing template, printed, faxed, or emailed.
	Electronic Lab Reporting	■	Incoming laboratory data can be captured from HL7 data feeds via the integrated HL7 engine. FreeMed is negotiating with Quest Diagnostics to build a bridge to retrieve electronic lab result data. Laboratory documents received as faxes can be stored in the document manager.
	Document Management	■	Stores all scanned documents and images in the DjVu digital document format. This format allows for high quality document images to be compressed to very small sizes. Scanned documents may be filed into various configurable document categories.
	Decision Support		FreeMed does not offer a general solution for integrated decision support.
	Disease Management	▒	FreeMed has an outcomes management module that focuses on episodes of care. It allows physicians to look at courses of disease across different patients. In addition, the application allows semantic tagging of conditions, which enables the classification and analysis of similar conditions. For example, conditions may be classified within certain diagnostic families (such as "pulmonary disorder"). These tags may then be used to build reports of patients with conditions of interest.

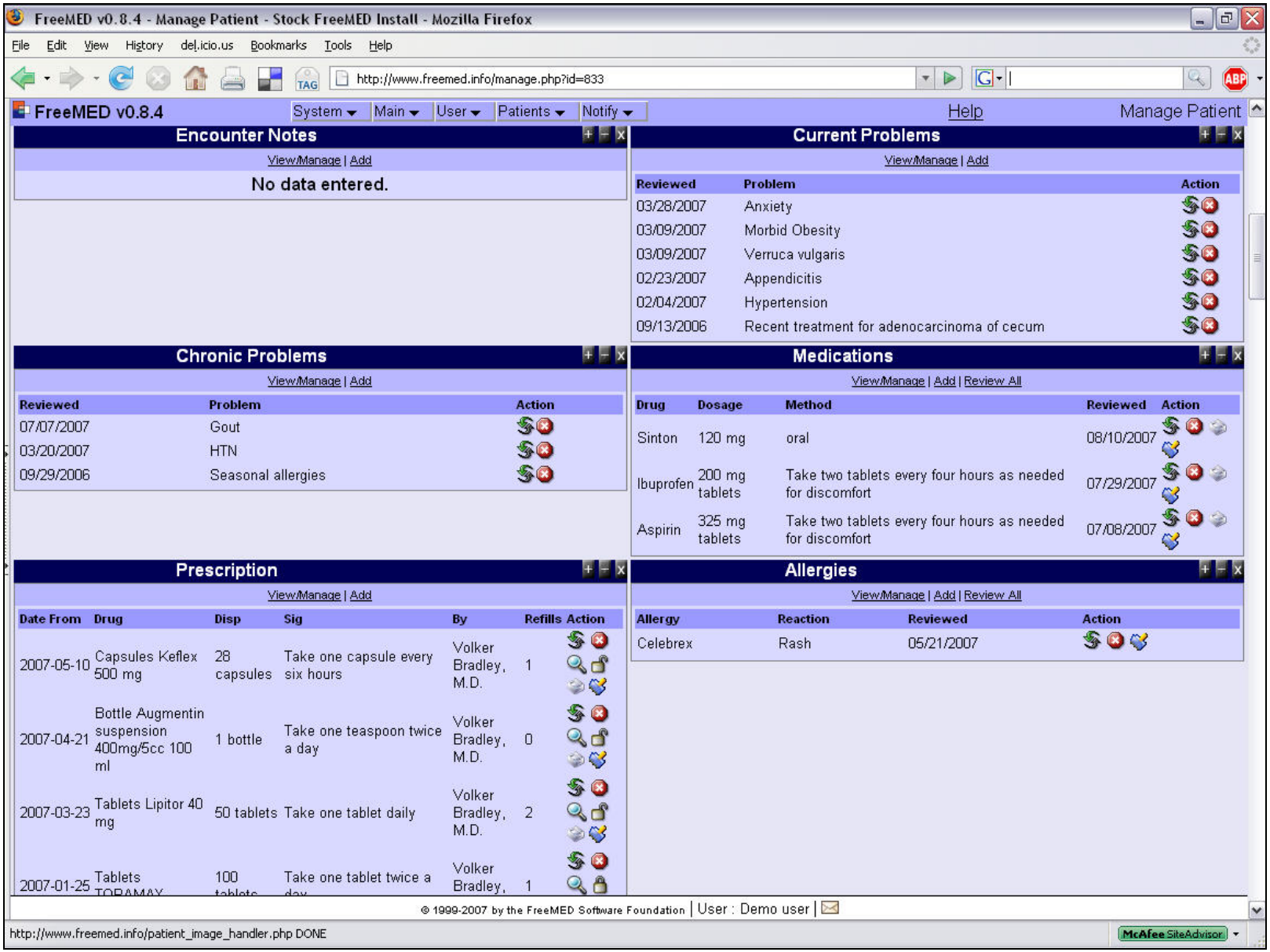
Legend: ■ Functionality Supported ▒ Some Functionality Supported □ Not Supported

Table 4 – FreeMed Functionality (2 of 2)

Area	Feature	Support	Support Details
General	Interoperability/ API		FreeMed has a native HL7 parser for receiving HL7 data, and it can also generate and receive data in the CCR format. In addition, a Mirth connector is available for integrating FreeMed with the Mirth Interface engine. The upcoming release will include a "data relay" system that will support API access to FreeMed data. The currently supported programming language bindings are Perl, Python, Java, C#, and PHP. The relay system will provide access control to patient data using the phpGACL system. Data is transmitted in the JSON format, an emerging data format similar to XML that is used in Web 2.0 applications.
	Standardized Vocabularies		FreeMed has support for ICD-9, LOINC, HL7, CCR and HCPCS. Some vocabularies, such as CPT4 and SNOMED cannot be used without additional licensing. As such, core functionality cannot be built around these coding systems. CPT4 and SNOMED can both be loaded into the system, but this must be done at the practice site after the necessary licenses have been obtained. FreeMed also has support for the HL7 OID database.
	Other		• HL7 Transcription Interface loads encounter data sent from transcription service into patient record. • Import of external progress notes from hospital system • Dictation interface

Legend: Functionality Supported Some Functionality Supported Not Supported

Figure 2 - Screen shot from FreeMed EHR system



Installation Options and Architecture: FreeMed is intended to be installed at the practice site on the practice's existing hardware. The FreeMed Foundation does not offer a remote hosting service at this time, although certain of the third-party support organizations are beginning to offer this option.

FreeMed is built on the common LAMP stack. Its architecture is distinctive in that it provides a pluggable module system that is self installing. Practice sites interested in adding one of the available FreeMed add-on modules can download the module from the FreeMed site and place the package into a modules directory. The system automatically detects a new module and integrates it into the existing system, adding any necessary database tables in the process. Modules may be just as easily removed. Example modules include a progress note template editor, a letter template editor, and an anesthesiology calendar. This approach has cost and convenience advantages for physician practices, because no programming experience is needed to incorporate new or optional features into the existing system.

Support Services: The FreeMed 501(c)3 does not itself offer implementation support services. Rather, there are a small number of licensed commercial partners that offer these services on behalf of FreeMed. These commercial companies provide all of the standard support services, including hardware and software installation, customizations, and practice site support. The commercial partners all must sign the FreeMed CLA that guarantees that code developed by them will be available to the FreeMed Foundation. In addition, the support firms provide the FreeMed foundation with 10% of their total gross revenues. These fees, in addition to charitable donations, are the only sources of funding for the FreeMed Foundation. In exchange, the FreeMed foundation gives their licensed commercial partners development support, third tier technical support, and prioritization with respect to bug fixes and feature enhancements.

There are currently two companies that are corporate partners of FreeMed Foundation, B-Mas, LLC (Jacksonville, FL), and UROWeb, LLC (Burlington, CT).

Each company offers a similar set of services to aid implementations of FreeMed. These corporate partners also contribute to the code-base to a minor degree. For example, B-Mas has produced a module for a behavioral health service customer that assists with the tracking of methadone treatment. The Corporate partner program is fairly new. B-Mas, has been formally supporting FreeMed for less than two years and has supported two practice sites. UROWeb is the newest member and is currently looking for clients.

Open Source License: The core architecture of the applications is entirely GPL. There are a few libraries that are distributed under LGPL, but FreeMed strives to distribute as much of its software under GPL as possible. This has prevented the inclusion of libraries and databases that could enhance functionality but are incompatible with the GPL license. FreeMed's reasoning behind this policy is that it will help to ensure that the software lives beyond the lifespan of the FreeMed organization, if necessary.

Development Community: Jeff Buchbinder, under the guidance of the FreeMed board of directors, provides the overwhelming majority of the code for FreeMed. There are limited contributions from the larger open source community, although public contributors who have signed a CLA and have demonstrated proficiency with the code are allowed to check code into the system in a highly constrained way. It is noteworthy that FreeMed has a larger international

following than the other FOSS EHRs this study reviewed, and the application has been localized in at least six different languages.

Analysis: FreeMed is the oldest of the FOSS EHR projects available today. The project shows a strong commitment to ensuring that its own code and the third-party libraries that it uses adhere strictly to open source licensing. However, in some cases this limits the solutions that FreeMed can offer to their user community, such as drug interaction checking (since no open-source drug interaction knowledge bases yet exist). FreeMed also realizes the importance of coded and normalized data to allow clinically useful reporting and analysis capabilities. FreeMed also has a strong approach to system interoperability with their data relay system.

Practices with the best chance for achieving successful implementations with FreeMed are small to medium practices that are sufficiently skilled to learn the system themselves or have an adequate information technology budgets to cover third-party support. The FreeMed project and its potential users would likely benefit from the development of a hosted version of the application.

Final Word: FreeMed is a good solution for small practices that can find effective third-party support firms or have significant technical capabilities themselves.

OpenEMR – Commercial Model

The OpenEMR Schism. Of all the FOSS EHR projects, OpenEMR has had the most colorful history. In 2001, the Synitech Corporation released a proprietary practice management system called Medical Practice Professional. A year later, the company decided to release the code base for this product under an open source license and a new name, OpenEMR. In 2004, a company named Pennington Firm (PennFirm) adopted the public OpenEMR code base and began to enhance and install the system under the commercial open source model. In 2005, PennFirm went out of business and the source code was, again, publicly posted to the SourceForge website, where it began to grow a small but loyal following under the community open source model. Some time after this, the former owner of PennFirm, Walt Pennington, sold the website and domain name of the project, openemr.net to the information technology consulting firm, Possibility Forge. At that point, Possibility Forge adopted the software and again began to enhance and install the system under the commercial open source model. Meanwhile, the community open-source project working with the code previously posted to SourceForge continued to operate, now with its own distinct code base. Hence, the OpenEMR project had forked into two distinct projects, which are now operating independently and gradually drifting apart. Because the rights to the OpenEMR trademark have not been established, both projects are currently using this name.

Given their distinct code bases and organizational models, this study reviewed the two projects separately. This study refers to the commercial OpenEMR project that is sponsored by Possibility Forge as OpenEMR(P), and the community OpenEMR project using the SourceForge website as OpenEMR(S). When this study refers to OpenEMR, both projects are implied.

Project Background: Possibility Forge is an enterprise software consulting firm with a focus on Java J2EE technologies. They offer a broad spectrum of consulting services to a wide variety of industries from health care to light manufacturing. Possibility Forge first became involved with the OpenEMR(P) project as a consulting firm that provided integration solutions for

OpenEMR(P) users. It has since become the principle contributor to the code base, taking over for the previous curator of the code, the now-defunct Pennington Firm. Possibility Forge has been working on its open source version of OpenEMR(P) since early 2006. The latest released version is v3.2.

Functionality: OpenEMR(P) offers a mature combination of practice management and EMR features. One of the strengths of the system is that Possibility Forge offers a number of modules that add functionality to the system that is unavailable in the other reviewed FOSS EHR projects. Of note is drug-interaction checking within the prescription ordering module, a module that leverages the full capabilities of the Cerner-Multum drug knowledge base (which requires separate licensing). Another such module provides coding assistance and billing pre-checks. Possibility Forge has also been very active in participating in the development of the NHIN interoperability demonstrations and has implemented a number of solutions stemming from this effort into the OpenEMR(P) product, including the IHE profiles, document store, and Eclipse OHF Bridge for SOAP-based interoperability with the application. The product lacks template-based note generation, a feature offered in certain of the other FOSS EHR projects. The functionalities listed in Table 5 and Table 6 represent the current release of the OpenEMR(P) project (v3.2). A screen shot of the application appears in Figure 3.

Table 5 – OpenEMR(P) Functionality (1 of 2)

Area	Feature	Support	Support Details
Practice Management	Scheduling/ Calendar	■	The calendar feature has multiple scheduling views with configurable scheduling categories.
	Patient Registration	■	The application offers a patient registration system with demographics, HIPAA preferences, employer, insurance and secondary insurance provider, medical history, and lifestyle sections.
	Billing System	■	OpenEMR(P) provides standard EDI services to transmit bills to billing services or directly to payers. Possibility Forge offers a transaction service to their hosted clients (available for a fee). Some decision support for billing exists in the application, but there are not a lot of rule sets available.
	Reporting	▒	A lot of the core administrative reports are implemented as HTML-based reports. Any standard report-writing application (e.g.., Crystal Reports) can be connected to the OpenEMR(P) database, but experience with these reporting tools and a strong knowledge of the database is needed to write *ad hoc* reports.
Electronic Medical Record	Problem Lists	■	Lists of current issues and active diagnoses are available.
	Note Generation	▒	SOAP-style notes may be created for each patient. These notes are free-text only, and templates are not currently supported.
	e-Prescribing	□	A SureScripts module is being developed for the next release of the product.
	Standard Prescribing	▒	Prescriptions and patient-reported medications may be recorded. Prescriptions may be printed, emailed, or faxed. A list of medications is provided by the Cerner Multum Lexicon. The prescription functionality may be extended to provide drug-interaction checking by integrating with the Cerner Multum knowledgebase, although there is an additional licensing cost to use this resource.
	Electronic Lab Orders	▒	Electronic ordering is not included in the core functionality of the system. Possibility Forge offers an NHIN module that allows for orders to be transmitted using the IBM OHF bridge.
	Standard Lab Orders	■	Laboratory orders may be entered into the system and printed.
	Electronic Lab Reporting	■	Laboratory data may be received electronically via a native HL7 interface or may be received as a fax and stored in a document repository.
	Document Management	■	A document management module is present. Possibility Forge has implemented the IHE document repository. An automated document scanning solution is available that uses barcodes to automatically attach a document to the appropriate patient's record.
	Decision Support	▒	Generic decision support is not available in the core system, although Drug/drug and drug/allergy interaction checking is available via the for-fee Cerner Multum prescription module. There is also a for-fee module that uses a 3rd party database to provide assistance for diagnoses and procedure coding.
	Disease Management	□	No general disease management features are currently available.

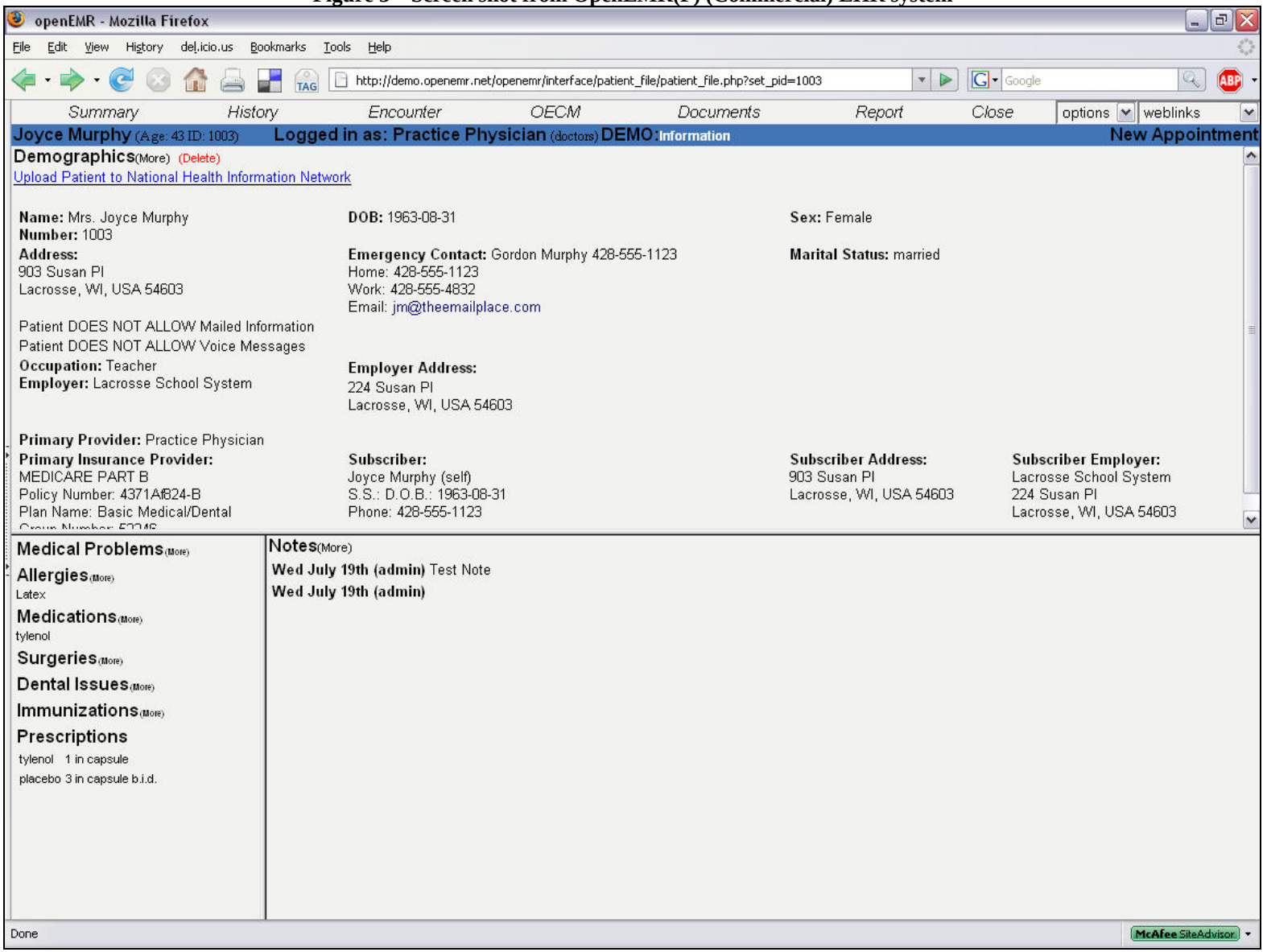
Legend: ■ Functionality Supported ▒ Some Functionality Supported □ Not Supported

Table 6 – OpenEMR(P) Functionality (2 of 2)

Area	Feature	Support	Support Details
General	Interoperability/ API		OpenEMR(P) uses an implementation of IHE profiles (the HL7 Clinical Document Architecture for the most part) for system interoperability. The application uses the Eclipse OHF bridge for access into the system via the SOAP protocol.
	Standardized Vocabularies		There is generic coding available in the core system for everything needed for billing in the U.S. (such as CPT4 and ICD-9). There is also a pluggable coding module available. This module can support any coding system and provides mapping capabilities to other coding systems as well as the internal representation of clinical concepts.
	Other		• IHE profiles interface • NHIN interface • SQL Ledger and other billing ledger modules

Legend: Functionality Supported Some Functionality Supported Not Supported

Figure 3 – Screen shot from OpenEMR(P) (Commercial) EHR system



Installation Options and Architecture: Currently, Possibility Forge has two installation options. Historically, OpenEMR was available only as an onsite installation. Since taking on the project, Possibility Forge has begun to offer a hosted ASP model for the application. Users of this service are charged an initial set-up fee plus a monthly subscription fee. Possibility Forge indicates that it has more practices on the hosted solution than any other installation option. Possibility Forge also offers a pre-built appliance that can be shipped to a practice site and integrated with the existing network.

The current version of OpenEMR(P) is built on the latest version of the LAMP stack. However, Possibility Forge is in the process of migrating the code from the current PHP-based architecture to an architecture based on Java J2EE. One of the main goals of migrating to J2EE is to add more robust security and interoperability options, as well as improved scalability for very large practice sites. Release of the next version of OpenEMR(P), which includes the migration to the Java-based architecture, is scheduled for Q1 of 2008.

Support Services: Possibility Forge provides a complete set of services for all types of practices, from solo practitioner offices to enterprise-wide installations. Small clinics can download the software and perform the installation themselves (if they are proficient with the technology) or they can contract directly with Possibility Forge for implementation support. In many cases, Possibility Forge can perform installations remotely. In large enterprises, Possibility Forge works with the local I.T. departments to develop an implementation plan and then implements this plan on a fixed-bid basis.

Although Possibility Forge is the primary provider of service and support for OpenEMR(P), Possibility Forge also has a number partners that provide some on-going support services. These partners are typically practice sites that have implemented OpenEMR(P) and are sufficiently familiar with the application and the code base to support other practice sites. These secondary support groups work in conjunction with Possibility Forge.

With the upcoming release of the Java version of the OpenEMR(P) project, Possibility Forge is completely overhauling the code base of the application. Possibility Forge intends to offer a migration path for their current PHP-based customers to the Java version of the server by providing an installer that helps to migrate the data. Furthermore, Possibility Forge states that the PHP and Java versions will continue to co-exist and both offer the same features for at least one year. Possibility Forge's plans for the PHP version of the application beyond this period are unclear.

Open Source License: The OpenEMR(P) software is available under GPL for the core system, and Possibility Forge has no plans to change this licensing model. At the same time, there are several third-party modules that are integrated with the core system and are offered under non-GPL licenses. For example, Possibility Forge leverages the Multum Medication database for use in the OpenEMR(P) prescription ordering module. This module is freely available for download, but practice sites must purchase and install the Multum drug interactions database to use it. The Multum database is proprietary and

therefore not available for download with the rest of the software under a GPL license. Although the source code for the prescription module is distributed under GPL, it is not functional without the proprietary drug database provided by Multum. Practice sites interested in this functionality may contract with Possibility Forge to obtain both the database and the prescription module, or they may download just the prescription module from Possibility Forge and purchase the Multum database separately from Cerner.

Possibility Forge offers a number of such external proprietary libraries and database solutions for the OpenEMR(P) product, and they are willing to provide non-GPL software where necessary to provide expanded functionality for their customers.

Development Community: As is typical with commercial organizations like Possibility Forge, development is handled mainly by its staff developers, of which there are between five and ten. There are very few contributors working outside of Possibility Forge, although there are a few OpenEMR(P) partners that contribute to specific modules used by Possibility Forge. These modules are not always published with the freely available OpenEMR(P) code, but may be obtained by contracting with Possibility Forge.

Analysis: Under the supervision of Possibility Forge, the OpenEMR(P) product is taking a new direction with a major migration to a Java-based architecture. Enterprise Java solutions are the primary business of Possibility Forge, so moving OpenEMR(P) to this new platform will bring their EHR product offering in-line with the rest of their business. It will also allow Possibility Forge to grow their business with larger practice site implementations. Possibility Forge has been focusing on support of the Nationwide Health information Network (NHIN) and Integrating the Healthcare Enterprise (IHE) interoperability profiles. The company was one of five members on the IBM NHIN architecture board and have been participating in the demonstration projects for the NHIN as part of the IBM consortium. Possibility Forge has taken a more pragmatic approach to module development, offering a wider variety of special licensed modules than the other FOSS EHR projects this study reviewed. These modules allow for extended application functionality, but are not fully GPL-compliant and often require extra licensing fees.

Possibility Forge is in the process of seeking CCHIT certification for its next release of OpenEMR(P).

Final Word: The current version of OpenEMR(P) is a mature and highly functional EHR and practice management system. The product is currently in a transition period as the system is migrated to the Java architecture. Small practices have done well with implementing OpenEMR(P) in the past, though with the change to the new architecture may come a shift in the market segment targeted by Possibility Forge. The transition to the J2EE architecture is a significant change for this product and will require further review once a release is publicly available. Furthermore, the future for the current PHP version of the application is unclear.

OpenEMR – Community Model

The OpenEMR Schism. As described under OpenEMR – Commercial Model, the OpenEMR project had forked into two distinct projects, which are now operating independently and gradually drifting apart. Because the rights to the OpenEMR trademark have not been established, both projects are currently using this name, but this study reviewed the two projects separately and provide an assessment of each. The community OpenEMR project using the SourceForge website will be referred to as OpenEMR(S). When the project is referred to as OpenEMR, both projects are implied.

Project Background: OpenEMR(S) has been hosted on the SourceForge website since 2005. The project is managed and maintained by a core team of volunteer project members that contribute to the source code. The two principal participants today are Samuel Bowen, M.D., and Rod Roark. Dr. Bowen is an internist who implemented OpenEMR(S) at his own practice in 2004. He runs the OpenEMR(S) community site, oemr.net, and is the chairman of Open Source Medical Software, a North Carolina nonprofit organization committed to the advancement of the OpenEMR(S) project. Rod Roark is the president of Sunset Systems, a consulting firm that supports FOSS implementations and has extensive experience in developing, implementing, and customizing OpenEMR(S).

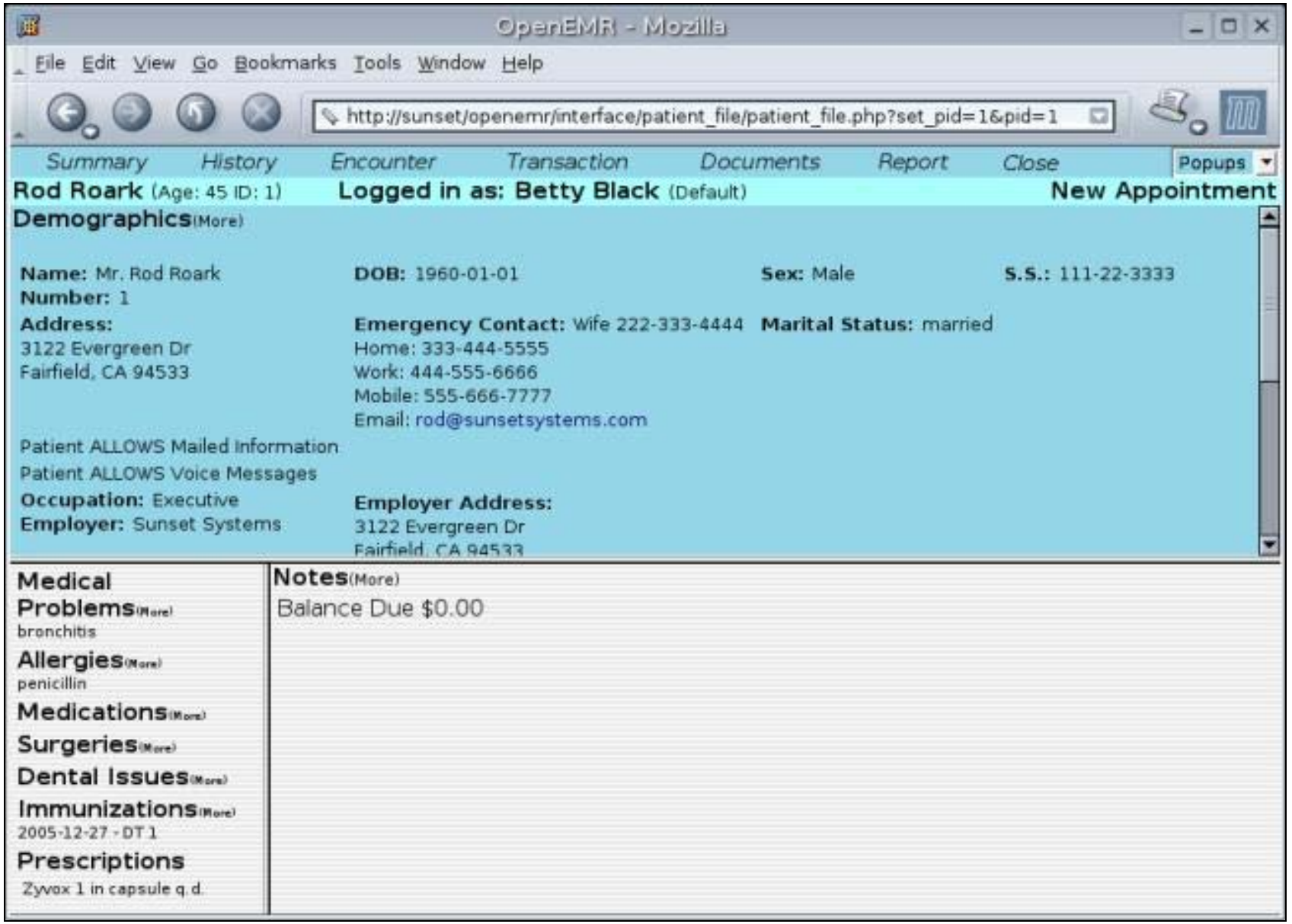
Functionality: OpenEMR(S) has a smaller set of features compared to other FOSS EHR projects reviewed. The scheduling and registration functionality is equivalent to that of OpenEMR(P), but the EMR functionality offers less support for external modules that provide advanced features (such as the drug interaction checking) than some of the commercial projects. The supported EHR features allow for standard encounter documentation but are very reliant on free-text entry with no template based-notes or coded problem lists. Notably, the application lacks laboratory ordering and lab order form printing capability. While a basic HL7 interface is present in the application, the support for HL7 import is limited at the moment. The features listed in Table 7 are those available in the 2.8.3 release of OpenEMR(S) (released August 2007). A screen shot of the application appears in Figure 4.

Table 7 – OpenEMR(S) Functionality

Area	Feature	Support	Support Details
Practice Management	Scheduling/ Calendar	■	The calendar feature has multiple scheduling views with configurable scheduling categories.
	Patient Registration	■	The application offers a patient registration system with demographics, HIPAA preferences, employer, insurance and secondary insurance provider, medical history, and lifestyle sections.
	Billing System	■	The application supports clearing house billing. A fee sheet template is available in the context of patient encounter. Providers may enter CPT4, ICD9, and HCPCS codes. Billing is integrated with the SQL Ledger application for running Balance Due functionality.
	Reporting	▒	A small number of administrative reports are included and presented in HTML format. Simple custom reports can be created from the database reporting section of reports menu.
Electronic Medical Record	Problem Lists	■	The application provides an active issue list that can list problems, allergies, medications, surgery, and dental items as separate issues in the list.
	Note Generation	▒	The SOAP Note structure is supported. Notes are free-text only with large text areas to capture notes. A dictation module is available to aid in populating the note fields.
	e-Prescribing		No e-prescribing is supported.
	Standard Prescribing	▒	Prescriptions may be recorded and printed, faxed, or emailed. The application pulls a medication list from rxlist.com. A list of the patient's preferred pharmacies is maintained.
	Electronic Lab Orders		No lab ordering is supported.
	Standard Lab Orders		No lab ordering is supported.
	Electronic Lab Reporting	▒	The HL7 Engine is rudimentary and untested. Scanned laboratory documents may be loaded into document manager
	Document Management	■	Document storage is available with configurable document categories.
	Decision Support		No general decision support or order entry checking is presently supported.
	Disease Management		No general disease management features are currently available.
General	Interoperability/ API		No formal system API exists for patient data interoperability. Patient data may be imported/exported in XML format but this feature is fully documented. One-off import/export solutions have been created for special purposes.
	Standardized Vocabularies	▒	Billing codes and diagnoses may be coded. (CPT, ICD, and HCPCS)

Legend: ■ Functionality Supported ▒ Some Functionality Supported □ Not Supported

Figure 4 – Screen shot from OpenEMR (Community) EHR system



Installation Options and Architecture: OpenEMR(S) may be installed on site or it may be remotely hosted. On-site installations are typically carried out by support vendors or technically experienced personnel at the practice sites. Users may download the latest release of the application directly from the SourceForge website. Instructions for download and server configuration are also freely available from oemr.org. The oemr.org website also lists vendors who provide on-site implementation support and hosting solutions. OpenEMR(S) is implemented on the latest version of the LAMP stack.

Support Services: Installation and support for OpenEMR(S) is offered by third-party vendors (listed in the Commercial Help area of the oemr.org wiki). Most of these vendors offer on-site implementation support services, and a few offer practice sites the option to buy servers preconfigured with OpenEMR(S) and billing components. In addition, certain firms (e.g., Sunset Systems) also offer a virtual machine for product-evaluation purposes. This software package can be run on any virtual machine player and allows users to try out the system using their own data in a secure environment prior to deciding to implement the full system. For example, this software allows one to run a virtual Linux operating system and the OpenEMR(S) web application on a Windows XP computer. This is essentially a software version of the appliance distribution mechanism, although recommended only for demonstrations and light use.

Open Source License: Everything that is in the SourceForge project repository is available under the GPL license. The project leaders are very dedicated to remaining completely open.

Development Community: There are currently eight developers associated with the project who are contributing code to some degree. In addition, there are between five and ten contributors that work with the registered project participants to submit patches to the code base. These participants contribute code on their own behalf and on the behalf of various practice sites that have implemented the EHR. These practice sites occasionally sponsor the development of additional features, and this model is the only real means for adding new functionality to the public code base.

Feature development and prioritization is solely driven by what practices want and are willing to fund. Rod Roark performs a lot of mediation with customers and other developers to ensure that redundant, inappropriate, or ineffectual features are not added. However, if a practice site wishes to sponsor the development of a feature, Rod Roark has indicted that he finds a way to get it done. If no one sponsors the development of a feature (even one desired by many users), that feature is not developed.

Analysis: OpenEMR(S) is an exclusively community run project. There is no formally defined project leadership or management structure, although a few members of the community have assumed these roles informally. Also, Dr. Bowen is in the process of establishing a 501(c)3 nonprofit organization to organize funding for and consolidate the management of the OpenEMR(S) project. Development is dependent on the willingness of practice sites to provide funding for development projects, and finding such practices has been a significant challenge for the project. This has resulted in a somewhat less robust EHR product, compared to the other FOSS projects this study reviewed.

Final Word: The OpenEMR(S) application is suitable for small practices with less demand for complex EHR functionality and having access to third-party support services (like those listed on the oemr.org website). Additionally, this project is the most open with respect to accepting contributions from individuals, and hence would be a good option for providers interested in contributing to an open source

EHR project. The project would greatly benefit from the establishment of a higher level management structure, such as the Open Source Medical Software foundation proposed by Dr. Bowen.

WorldVistA EHR

Background. WorldVistA EHR is based on a public domain version of VistA, the clinical information system in use at Veterans Health Administration facilities around the world. The source code for this version (known as FOIA-VistA) was placed in the public domain pursuant to a freedom-of-information-act (FOIA) request. In 2005-2006, the Centers for Medicare and Medicaid Services (CMS) adapted VistA for use in ambulatory practices and community clinics in the private sector under the name VistA Office EHR (VOE). VOE was intended as a low-cost alternative for practices and clinics seeking EHRs. The goal of the software modifications was to remove extraneous elements of VistA that were not relevant in the private sector (e.g., classification of combat injuries), to add missing elements to VistA that were not needed in the V.A. system (e.g., support for pediatric care), and to implement functionality necessary to meet the ambulatory EHR certification criteria of the Certification Commission for Health Information Technology (CCHIT). Additional changes to the code base simplified software maintenance and future enhancement.

VOE included extensive clinical functionality and some practice-management functionality. In 2006, CMS sponsored a beta test of VOE at ten practice sites, with management of the code base and technical support provided by the nonprofit firm WorldVistA. The testing indicated that the system provided value in a number of areas, but that successful adoption required extensive content customization, effective end-user training, and interfaces to lab and practice-management systems. A detailed evaluation of the beta test is available at http://www.sujansky.com/vista.php.

After completion of CMS's VOE project in March, 2007, WorldVistA continued to manage the code base and to make additional enhancements, renaming the resulting system WorldVistA EHR. In May, 2007, WorldVistA EHR achieved CCHIT certification, the first open source ambulatory EHR to do so.

Given the extensive evaluation of the VOE system that was conducted in 2006 (cited above), further research was not performed on WorldVistA EHR for this study. Readers interested in specific changes to the system since the earlier evaluation are referred to WorldVistA (http://www.worldvista.org/World_VistA_EHR.