Secure Development of Healthcare Medical Billing Software

Paige Peck
Department of Computer Science
College of Charleston
Charleston, SC, United States of America
e-mail: paigepeck@hotmail.com

Aspen Olmsted
Department of Computer Science
College of Charleston
Charleston, SC, United States of America
e-mail: olmsteda@cofc.edu

Abstract— **Healthcare medical billing has been progressing into the digital era for several years, but it has been a slow and expensive process that has left many parts of the industry behind. One of the many things that have been overlooked in the progression is security, especially now that medical records are worth far more than credit card numbers on the black market. Another issue the healthcare industry has been dealing with is the lack of systems being incorporated. Currently, there are companies that are using printed out spreadsheets to find rules for coverage of a procedure based on any insurance company's policies. Using business rules engines and rule validations, we make it easier for a doctor or office to type in lab results and see whether a procedure will be covered by a patient's insurance company. We chose to create these using the Salesforce Cloud development service.**

Keywords- Healthcare billing software; Current Procedural Terminology; CPT codes; Healthcare Common Procedure Coding System; HCPCS codes; Salesforce Cloud development

I. INTRODUCTION

Healthcare has been transitioning to digital for years now, but up until recently, it has been a slow process. As Ballas [1] discussed back in 2001, the Internet would help to reduce the ever-rising costs of healthcare and would give the patient more power by allowing them to become more educated about specific medical procedures. He goes on to suggest that the internet will be able to help medical record keeping by giving access to these files on the web. While some of these have been implemented using Cloud development such as CureMD, Practice Fusion, and Athenahealth [2], healthcare is still behind where it should be. Payor rules are still being viewed on printed excel spreadsheets to find the information needed, and doctors do not have easy access to them electronically in a simple application. Having this could prevent doctors from prescribing drugs or procedures that should not be due to the patient's needs.

Now over fifteen years later, the conversion of healthcare to the Cloud is advancing, according to Ratchinsky [3]. While $3.73 billion was spent on Cloud services for healthcare in 2015, that number is expected to rise to $9.5 billion by 2020 [3]. Healthcare is moving towards the Cloud Technology more because Cloud applications are so flexible with scaling, are highly accessible, and are cost effective. Within a few years, it is expected that there will be less direct face-to-face interaction between patients and their providers [3]. Not only will the patient have more access and control of their medical records, but the use of business rules engines will help ensure that someone cannot be automatically prescribed a drug or procedure they cannot have without their knowledge. Business rules engines can be set up by healthcare providers and administrators using near English formats for non-software developers to easily set conditions on anything that a patient could automatically access. Having a system where admins could set these rules would save countless amounts of dollars. It would also prevent errors from occurring that could lead to a patient having a treatment they should not be able to have due to health conditions.

But is the Cloud secure enough for the many different laws concerning healthcare privacy? Guccione [4] discusses this very question along with a recent break into an Indiana-based medical software company. According to the company, patient names, email addresses, Social Security numbers (SSNs), and medical records were possibly stolen. The criminals also managed to break in the company's Cloud service, a system which allowed the patients to gain access to their medical records remotely. Healthcare is being targeted more with medical records having an increased value on the black market, far exceeding credit card numbers by tenfold. However, with all these break-ins and loss of data, at the time of the article's writing there had not been an update to the Health Insurance Portability and Accountability Act (HIPAA) rules on Cloud services in over three years [4]. With Cloud computing on the rise and healthcare using it now more than ever, the security regulations will need to be updated far more frequently. Those creating the applications will also need to consider potential outside threats.

To ensure that a system is secure, we put an effort into Confidentiality, Integrity, and Availability (CIA). Most research about healthcare security focuses on the Confidentiality of the system due to the nature of the data that is being stored and used. As the system will be interacting with personal health information, it is important for the system to keep the records confidential and secure. We are also working on the Integrity of the billing required by the clients. Most physicians have said that Integrity is the most important aspect of their job in the medical field. Integrity is also in the HIPAA Security Rules by stating that one must "implement policies and procedures to protect electronic Protected Health Information (ePHI) from improper alterations or destruction" [5]. Because the application is designed in Salesforce, availability is based mostly on their platform. Salesforce has several data centers spread across the United States in case of

 ISBN: 978-1-61208-582-1

power failure, network connection issues, or hardware failure. Because of these centers, the loss of data is very minimal, measuring at mere seconds of lost data while the other centers take the traffic from the failing center.

The organization of the paper is as follows: Section II describes the related work and how others have attempted to tackle the mentioned issues. In Section III, we give a motivating example and describe a rule, why it is enforced, and why it is important. In Section IV, we go over the implementation of what we are building, why we chose a building in the Cloud, and show how we got to the point we are at. Section V discusses the results of our work such as what was good, bad, and difficult. We finish off the paper with Section VI that goes over the conclusion and future research.

II. RELATED WORK

There are several ways to ensure the correctness in healthcare medical billing software. One of these ways is by using a business rule engine. These are functions that can be used to create business rules without the need of a programmer. Olmsted and Stalvey [6] discussed these business rule engines and how they have been designed to allow users and non-programmers to change the business rules without changing the application code. According to their research, ninety percent of people completing a survey from International Data Corporation in 2007 said they change their business rules at least annually, if not more frequently. Of those that change, thirty-four percent change the rules monthly. There are several methods on how to develop these rules based engines, such as Drools [7]. Drools is a business rules management system. Drools facilitates the definition and enforcement of business rules engines. Another process was created and implemented by Abdullah, Sawar, and Ahmed [8] using Structured Query Language (SQL) specifically for applying billing compliance rules on medical claims. Medical billing is very complex and ever changing. Many times claims are rejected initially causing payments for services rendered to take a long time. Using the MTBC Rule Based System makes it easy for a user to edit rules in near English format, which is then translated into SQL statements. This system is currently being used by billing executives to enter medical claims into the database. The system is being continually updated. One of these newer updates is an "Auto Rule Generator" based on machine learning techniques [8].

Due to privacy laws dealing with medical information, security is an imperative component when designing medical billing software. Löhr, Sadeghi, and Winandy [9] discuss the lack of security in current online healthcare software and possible solutions to these security flaws. Throughout their paper, they describe the different types of electronic healthcare options giving several examples as to why it is not secure and how the systems can be breached. From there, they discuss the solution by separating medical data from billing and accounting data using a working prototype called Trusted Virtual Domains. They are also creating a user interface for this prototype. Though they have solutions to several of the issues they bring up, there are still a few security concerns involving these solutions. They discuss some of these such as the use of USB sticks that could be carrying malware and viruses.

Another solution to the risks of healthcare systems online is discussed by Kobayashi [10] by using Open Source Software (OSS). The use of open source software is also a solution to the rising costs of healthcare software. OSS is developed by volunteers and is provided 'as is' usually, which makes people skeptical about the security of the product. However, evaluations have been done on proprietary software that shows OSS has often been more reliable and has fewer bugs in the source code. OSS has also been shown to release patches more often that fix identified vulnerabilities.

Vanitha, Narasimha, and Chaitra [11] discuss using Electronic Health Records (EHR) and electronic billing systems on the Cloud with the platform MedBook. MedBook uses open source Cloud computing to help fight rising costs and detect fraudulent activities in the healthcare system. They continue to discuss how Cloud computing allows for costs being reduced when using this infrastructure. Reliability is improved when redundant sites are used, and security is improved because of the centralization of data and resources that focus on security. MedBook is a Software-as-a-Service (SaaS) application [11]. This is like our application since it is utilizing Salesforce, which is considered both a SaaS and a Platform-as-a-Service (PaaS). Software-as-a-Service is software that is hosted in the Cloud, which allows users to access the application through a web browser or an application on PCs or mobile devices.

Begum, Bhargavi, and Rani [12] wrote a review on how healthcare was utilizing the Cloud. This article discussed how organizations are still using paper records and handwritten notes to pass around data and come to conclusions. The authors go on to discuss possible solutions to potential problems when using the Cloud for medical data and the benefits that would be seen such as preventing any Protected Health Info (PHI) from being stored on hospital computers. This would prevent the current PHI violations that have been occurring due to the theft of computers.

As stated earlier, Salesforce is considered a PaaS Cloud-based system, which allows the developer of the software to not worry about the operating system the platform runs on [13]. Olmsted and Fulford [14] discuss the problems with the costs of development with PaaS Cloud systems. They continue to discuss PaaS systems and group them into two categories, each of which is used in our system. The first one is the previously mentioned rule engine to check business rules that are often changing and should not be coded directly into the system. The other being an importing feature using Comma Separated Values (CSV) formats and state that this should be validated to ensure that the database is secure.

III. MOTIVATING EXAMPLE

We are contributing to this industry by creating an application for a healthcare consultant agency. This application is being developed using the Cloud development

 ISBN: 978-1-61208-582-1

tool Salesforce. The focus of the application is on the lookup field for rules set for drugs, Current Procedural Terminology (CPT) codes, and Healthcare Common Procedure Coding System (HCPCS) codes. CPT codes are medical codes that are used to describe any medical procedure done by a healthcare provider. They are created and maintained by the American Medical Association. There are thousands of these codes split up into categories for medical coders to enter so the healthcare providers will be reimbursed from the insurance companies. Some of these CPT codes are variations of other procedures. These codes need to be entered properly, with the more specific variation chosen when possible, or a claim can be rejected due to the procedure not being covered. HCPCS codes were created by the Centers for Medicare and Medicaid. HCPCS codes are very similar to CPT codes, often the exact same, but they are used to represent Medicare, Medicaid, and other third-party payors. HCPCS codes are also used more as a specific drug where CPT codes are procedures done on a patient. The level II category HCPCS codes vary from the CPT codes in that they begin with an alphanumeric letter. The codes we use as an example fall under this category and begin with the letter 'J.' J-codes are the most common codes, and they are codes for non-oral medication and chemotherapy drugs that cannot be self-administered. HCPCS codes have more specificity than CPT codes, which includes many variations of equipment and drugs, so it is far more important for medical coders to put in the claims [15].

The drug we will use as an example is PROCRIT (HCPCS: J0885). According to the company that sells PROCRIT, "PROCRIT (epoetin alfa) [16] is used to treat a lower than a normal number of red blood cells (anemia) caused by chronic kidney disease in patients on dialysis and not on dialysis. Chemotherapy that will be used for at least 2 months after starting PROCRIT. A medicine called zidovudine (AZT) used to treat HIV infection" [16]. Every one of these drugs and procedures has requirements based on the patient's lab results. The requirements placed on the drugs and procedures are laid out by the insurance companies, or "payors." Because of this, a drug can have requirements from one payor that are not listed from a different payor. As an example, Medicaid might list a requirement for a patient's hemoglobin (Hgb) level to be below 10 to allow administration of PROCRIT. In contrast, BlueCross BlueShield might have a requirement for the patient's Hgb levels to be below 12 or not even have a requirement for the Hgb levels to allow administration of PROCRIT.

The rules placed on these drugs and procedures are what can cause a patient's claim to be rejected by the payor if the rules are not followed. With all these details placed upon a drug/procedure, claims are often rejected at first. Currently, the client is traveling and passing out laminated cards of these rules for the drugs. By doing this, they have cut down

claim rejections by nearly 50%. By creating this application, we will be cutting down far more claim rejections by making the rules a validation step when entering the values into the system in addition to displaying the rules of each drug/procedure based on the payor. This will save the healthcare industry thousands of dollars by ensuring the billers will get paid by performing the correct procedure on a patient. Table I shows the requirements for the drug PROCRIT. Hgb levels and hematocrit (HCT) are considered "OR" statements provided in the table. As an example, for "Anemia: Chemo Induced – Encounter for chemotherapy" the Hgb levels must be below 10 OR HCT must be below 30 for PROCRIT to be administered and covered. A disclaimer is also included on these laminated cards based on the drug. These disclaimers are a non-payable list of diagnoses that are not covered. An example of one of these non-payable diagnoses is "any anemia in cancer or cancer treatment due to iron deficiency." For the

TABLE I. PROCRIT J0885 REQUIREMENTS CHECK LIST

Rules	Anemia: Chemo Induced – Encounter for chemotherapy	Anemia: Chemo Induced – Encounter for chemotherapy	Myelodysplastic Syndrome (MDS) – No Secondary Requirements	MDS – Anemia in other chronic diseases classified elsewhere	Chronic Kidney Disease (CKD) – Anemia in CKD
Hgb for initiation	< 10		< 10		< 10
HCT for initiation	< 30		< 30		< 30
Hgb for continuation of Therapy		< 10		< 11	< 11
HCT for continuation of Therapy		< 30		< 33	< 33
TSAT	> 20%			> 20%	> 20%
Ferritin	> 80 ng/mL			> 80 ng/mL	> 80 ng/mL
Timing of Labs	Within 7 days Prior to Initiation of Therapy	Every 4 weeks for continued Therapy	48 Hours Prior to Initiation of Therapy	Every 4 weeks for continued Therapy	Within 7 days prior to Initiation of Therapy Every 4 weeks for continued Therapy

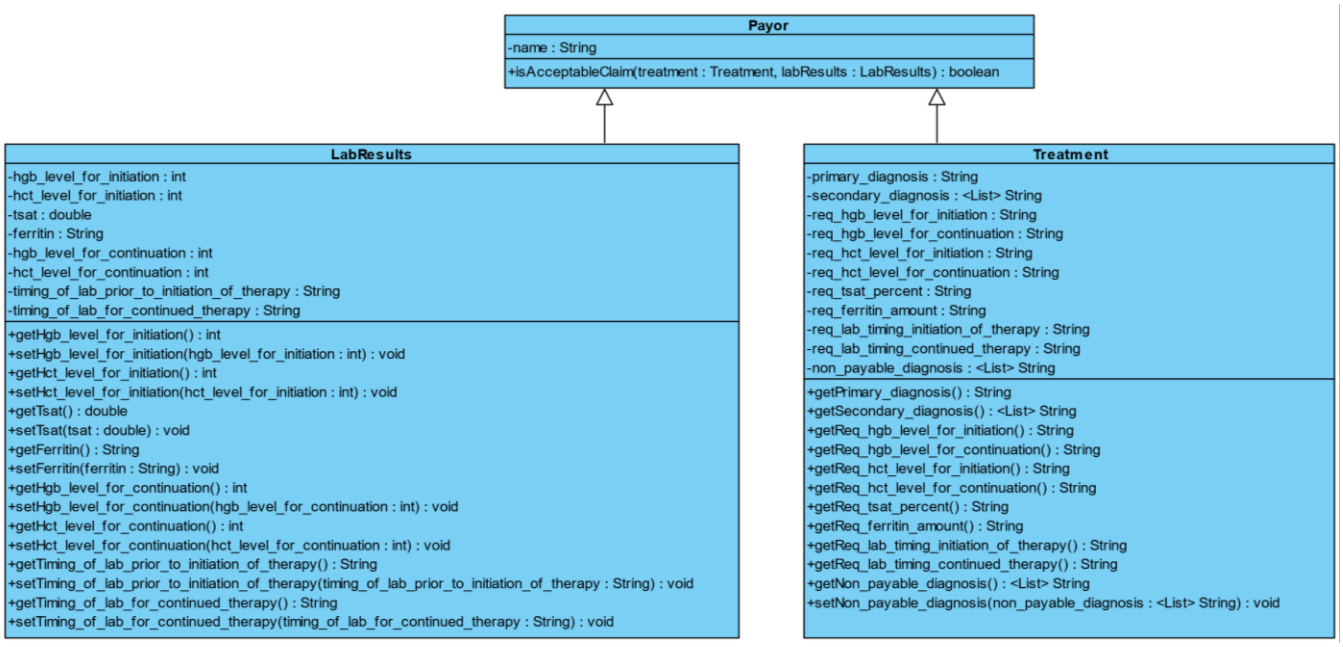

Figure 1. UML Class Diagram

application, the client requested that these disclaimers merely be displayed at the bottom of each drug. Figure 1 is a UML class diagram showing how the lab results and treatment requirements relate to the payor. In the system, a payor will accept a claim if the lab results are within range of the treatment requirements. The treatment variables are set as string variables due to the fact most of them include a comparison operator to check with the lab result variable.

To keep these requirements checked and ensure there is no error on the user's side, we are using the business rules engines mentioned previously. Salesforce has its own form of a business rules engine called "Validation Rules." These validation rules can be set on each object in Salesforce. The rules have functions such as "AND', "OR," "CONTAINS," and much more that can be used to validate a field or multiple fields of an object. After assigning the fields, operators, and functions, we can hit the "check syntax" button to make sure we typed everything in correctly. After assigning this rule, we can set the error message that will be shown when an error condition occurs. For the example drug PROCRIT, we can have an object called "Drug" with the above fields stored into the object's fields based on a lookup field for another object called "Payor." For one validation rule, we can read the lab results and parse through the text using "CONTAINS" to find Hgb or HTC levels. If we have found them, we ensure the level of the patient is below the values listed in Table I using the less than or greater than operators. If they are, the system continues down the list of validation rules set. Otherwise, it throws an error showing the user that PROCRIT is not covered under the selected payor.

IV. IMPLEMENTATION

The client first brought their idea for this application to our attention by stating that current insurance companies and healthcare organizations are searching through printed out Excel spreadsheets and finding the rules laid out by each payor for a specific treatment. These rules are not laid out in any easily trackable system. These rules need to allow for a transfer of information from the administrators who create and edit the rules to the doctors' offices. The offices need to be able to explain why a claim was not covered or a specific drug cannot be administered. As Begum, Bhargavi and Rani [12] discussed, the lack of proper healthcare applications on the Cloud, or even in general, is costing the industry millions of dollars. Figure 2 shows a model they created for a PaaS system where users can have a local electronic medical record and not have to manage the system framework. The application we are building is utilizing one of the PaaS development models, Salesforce.

The choice to use Salesforce for this application was simple as the client wanted the application to be created quickly and with a service that can be used on more than just a computer, such as use on a tablet or mobile device. Salesforce excels in both. It is also reliable and has very good support. Salesforce was also good because it is not too costly for the client's planned model. When it comes to security, Salesforce stays on top of current malware, phishing, and intrusion attempts and is constantly updating their system to reflect these. They have event monitoring, which gives a client detailed information about any action that is taking place on the system. They also use the most up-to-date authentication and encryption methods and hosts its data on a secure server environment [14].

To fix the clients first problem, they asked for an administration toolkit where those who used it could import and export rules based on Comma Separated Values (CSV) files. In these import and export pages, they requested an easy way to edit the rules and add new ones when needed. When

 ISBN: 978-1-61208-582-1

Figure 2. Platform-as-a-Service Healthcare Model

designing the system, we decided that two separate pages were called for, one for importing and adding rules, and another for exporting and editing the rules. We were given an excel spreadsheet from an insurance company as a template, and we modeled the system around this. For the importing function, if the file is a CSV and follows along with a given template that a user can download, they can easily import new rules after setting an effective date. Exporting works about the same way, where a user selects the fields they want to export to CSV, and the system then downloads a file with the rules selected under the fields that were searched.

The client was happy with the admin toolkit design and wanted us to move on to their next step before they planned to present the product as an early prototype to the insurance companies that they are consultants for. They wanted the next part of the application to focus on the doctors and offices that will use it, focusing on the specific rules of the drugs laid out by the payors. These rules are what we will be using to ensure the correctness of the software. The final product for this part of the application will allow a doctor to traverse through it on a tablet and enter the procedure with the constraints given, and the application will inform the doctor whether the drug can be administered or not.

V. RESULTS AND DISCUSSION

Throughout the process of creating this application, we discussed several options on how to handle validating the requirements for the drugs. At first, we discussed writing a parser and regular expressions to ensure the requirements were met. As this could potentially take some time to write and improve, we looked elsewhere to see if there was a better and faster way of doing it. We debated using business rules engines such as Drools, which was less complicated than parsing and using regular expressions but still not exactly what we were looking for. Creating a tool for a user to create these rules themselves was another option that has been done before, but this can create problems overall if a user mistakenly writes an incorrect validation. Then we came across the out-of-the-box rules validation that Salesforce controls for each object type. This built-in feature was already designed into a system we had been working on for over a year as well. If we write the validation rules properly, then this feature will do the work for us.

The difficulty lies in writing these validation rules properly to ensure they do the work correctly. As stated before, several of the drug requirements can have an "OR" associated with them, for example, HCT and Hgb levels each have their own requirement levels, but only one is needed to meet this. An example of a properly written validation rule based on Table I would say: "(Hgb for initiation < 10) OR (HCT for initiation) < 30". If one of these are true for an attempt of administering PROCRIT for anemia due to Chronic Kidney Disease or Chemo Induced, then the system will allow the user to continue. There are requirements for initiation of taking the drug and separate requirements for continuation of taking the drug that can be misunderstood or improperly set. As this is the main function we want doctors and users to trust, it will have to be very carefully checked that the validation rules entered are correct.

The next phase will be working closely with the client to build these validation rules ourselves, so the user will not be writing them. As each drug and payor combination has their own set of requirements, this will take some time to get all of them working. For now, we will be building out the rules that the client deems worthy for showing off a prototype to potential buyers. As they pass along the laminated cards, they will be showing off the application as an easier and all around better way for checking these requirements. Because it will be checking the data entered from the lab results, it will be easier for them to see when a drug or procedure will not be covered. administered or not.

VI. CONCLUSION AND FURTHER RESEARCH

In this work, we discuss ways to guarantee medical billing software to be secure on the Cloud and accurate. As healthcare makes the transition more to the Cloud, accurate and secure data is pertinent for the application if we want the clients to trust using it. There are several options one can use to ensure the correctness of the data entered, and the choice that is easiest to implement and follow is the one that is built in for us already using Salesforce validation rules. Salesforce is constantly monitoring new attempts at malware and phishing to give us one of the most secure Cloud development tools on the market. Future work will help us broaden the application for more users to access it and be able to easily add the ever changing and new requirements from the healthcare industry.

 ISBN: 978-1-61208-582-1

REFERENCES

[1] M. S. Ballas, "The Impact of the Internet on the Healthcare Industry: A Close Look at the Doctor-Patient Relationship, the Electronic Medical Record, and the Medical Billing Process," *Einstein Quarterly Journal of Biology and Medicine*, vol. 18, no. 2, pp. 79-83, 2001.

[2] R. Lowes, "Medscape," 15 May 2012. [Online]. Available: http://www.medscape.com/viewarticle/763894. [Accessed 14 August 2017].

[3] K. Ratchinsky, "Why the healthcare industry's move to cloud computing is accelerating," 27 June 2016. [Online]. Available: https://www.cloudcomputing-news.net/news/2016/jun/27/why-healthcare-industrys-move-cloud-computing-accelerating/. [Accessed 3 June 2017].

[4] D. Guccione, "Healthcare Informatics Institute," 20 July 2015. [Online]. Available: https://www.healthcare-informatics.com/article/cloud-safe-healthcare. [Accessed 05 June 2017].

[5] H. C. Pros, "Integrity: More than Just a Piece of the Healthcare Compliance Puzzle," 17 June 2014. [Online]. Available: http://www.healthcarecompliancepros.com/blog/integrity-more-than-just-a-piece-of-the-healthcare-compliance-puzzle-2/. [Accessed 28 June 2017].

[6] A. Olmsted and R. Stalvey, "Highly available, consistent, business rule filters," *The 9th International Conference for Internet Technology and Secured Transactions (ICITST-2014)*, 2014.

[7] "Drools," Red Hat, Inc., [Online]. Available: https://www.drools.org/. [Accessed 14 August 2017].

[8] U. Abdullah, M. J. Sawar and A. Ahmed, "Design of a rule based system using Structured Query Language," *2009 Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing*, pp. 223-228, 2009.

[9] H. Löhr, A.-R. Sadeghi and M. Winandy, "Securing the e-Health Cloud," *Proceedings of the ACM international conference on Health informatics - IHI '10*, 2010.

[10] S. Kobayashi, "Open Source Software Development on Medical Domain," InTech, 2012.

[11] T. N. Vanitha, M. Narasimha Murthy and B. Chaitra, "E-Healthcare Billing and Record Management Information System using Android with Cloud," *IOSR Journal of Computer Engineering (IOSR-JCE*, vol. 11, no. 4, pp. 13-19, 2013.

[12] F. Begum, K. Bhargavi and T. Suneetha Rani, "A Review on Healthcare in Cloud," *IJSTE - International Journal of Science Technology & Engineering*, vol. 2, no. 06, pp. 124-129, 2015.

[13] "Salesforce," Salesforce.com, Inc., [Online]. Available: https://www.salesforce.com/ap/. [Accessed 14 August 2017].

[14] A. Olmsted and K. Fulford, "Platform As A Service Effort Reduction," *CLOUD COMPUTING 2017, The Eighth International Conference on Cloud Computing, GRIDs, and Virtualization*, pp. 60 - 65, 2017.

[15] "Medical Billing & Coding Certification," [Online]. Available: http://www.medicalbillingandcoding.org. [Accessed 08 June 2017].

[16] Janssen Products, "Procrit epoetin alfa," Janssen Products, 24 July 2015. [Online]. Available: https://www.procrit.com/. [Accessed 09 June 2017].

 ISBN: 978-1-61208-582-1