MyHealthPortal Project

Ezequiel Romero
Department of Computer Science
rome9162@vandals.uidaho.edu

Adam Odell
Department of Computer Science
odel6278@vandals.uidaho.edu

Jamil Hasan
Department of Computer Science
jamil@uidaho.edu

ABSTRACT

The MyHealthPortal project is a healthcare website to facilitate the exchange of services between patients, doctors, insurance providers, and pharmacies. It operates on a database with patient-centric data with an internal record of different kinds of medications and medical services that can be exchanged. The website has a different interface for patients, doctors, pharmacies, and insurance companies, allowing all of them to access only the data they need and exchange necessary information. Patients can choose insurance plans, make appointments with doctors, receive diagnoses, order prescriptions, and view bills. MyHealthPortal is limited in not maintaining billing history after a patient switches health insurance plan. It also does not provide a method for cancelling appointments or orders. Future improvements may include only allowing patients to buy over the counter drugs without a prescription, allowing different pharmacies to carry the same drugs under different brand names, and allowing patients to decide between picking up a prescription or having the prescription mailed to them.

1 INTRODUCTION

"Websites have become the most important public communication portal for most, if not all, businesses and organizations" [2]. The nature of web design may seem very subjective, since proper design has become a critical element that can engage website and mobile application users; however, once broken into it's simplest forms, it is clear that all websites follow the same structure. All websites have a frontend, in which users can explore the website and all it has to offer. They can manipulate SQL tables and query all of the data in the database. All websites also have backends, in which the actual data is stored and manipulated. Here, web developers are able to mold their websites into a productive programs. They are able to use all of the data provided by the users to determine their certain access rights, privileges, and custom information. These users don't only have to be common people who use the cite regularly. Another important user is the administrator or administrator(s). There can be many types of administrators, but for the sake of brevity, lets take into account a single administrator of a website. Unlike the web developer, the administrator is in charge of both reviewing the common user's information as well as manipulating it, if necessary. The administrator is also given permissions by the web developer, who is also responsible for creating relations between administrators and users. Under a healthcare setting, however, these roles become much more complex. First, we have a patient who is able to access few items. Next we have service providers, who act as our administrators. These range from doctors, to pharmacy heads, to insurance companies. With these users in mind, we are tasked to develop a web interface sufficient for all of these users, as well as develop a proper MySQL database featuring all of their information and relationships. This schema is what will be discussed in the following report. There were a few reasons why we chose to develop a patient healthcare portal instead of a tax website. First was obviously because we had dealt with the healthcare schema previously through Hasan Jamil's lectures and assignments. Other than that, another good motivation for this project was its linearity. Unlike the tax project, this assignment seemed to deal more with the websites frontend.

1.1 Project Proposal

In this MyHealthPortal Project, my partner and I have created a healthcare website for patients, service providers, pharmacies, and insurance companies all through the same system. This follows the SNOMED-CT, patient-centric health care management format, in which patients are able to schedule appointments, purchase pharmaceuticals, and view billings from the provided service providers, insurance plans, and pharmacies. SNOMED-CT is a systematically organized computer process-able collection of medical terms providing codes, terms, synonyms and definitions used in clinical documentation and reporting [1]. With the structure and frontend in

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

place, the following will elaborate on the specific tables we have included into our backend database. In total, we have 16 tables, each chock-full of dummy data given by our Python script *Database-Generator* [3].

2 DESIGN AND ARCHITECTURE

My Health Portal is a health insurance portal for patients, pharmacies, doctors, and insurance companies to access healthcare information, request drugs and treatment, and compile billing information. The database schema revolves around patient data, primarily contained in the Patients, Membership, and ServiceRecords tables.

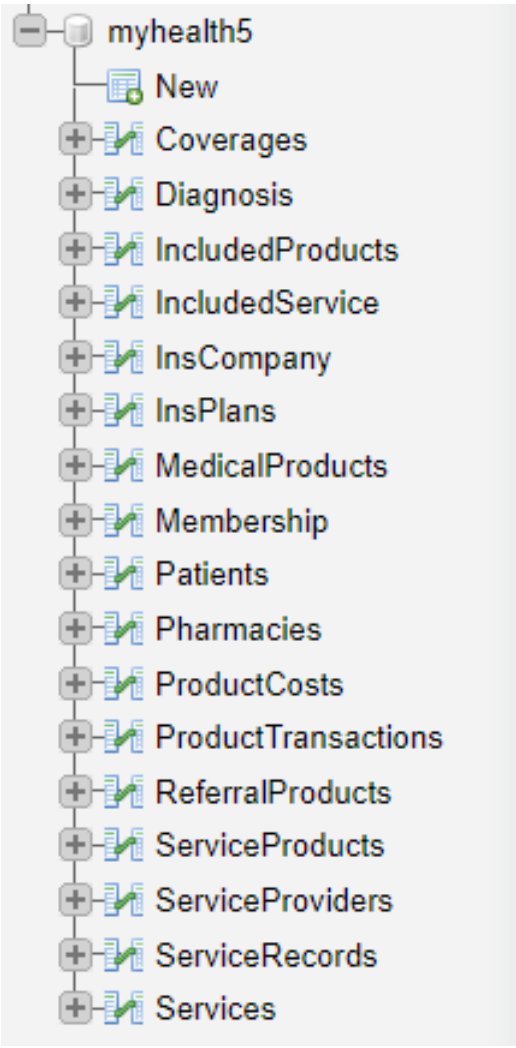

Figure 1: List of tables in database

2.1 Patients

The Patients table contains information on patients such as an ID number, contact information, and authentication information. Membership contains information on what insurance plans patients are covered by. ServiceRecords contains a history of the medical services that patients have received from service providers, such as doctors and hospitals.

PID	SSN	Password	Name	Address	DoB	Phone	Email
10000055	123456789	b59c67bf196a4758191e42f76670ceba	Hasan Jamil	247 Baker St	19600101	7349288944	hm_jamil@yahoo.com
10000054	123457686	5f4dcc3b5aa765d61d8327deb882cf99	Ezequiel Romero	901 Paradise Creek Street	20011217	2087775236	ezequielxromero@gmail.com
10000053	293842930	5f4dcc3b5aa765d61d8327deb882cf99	Ezequiel Romero	609 E 21ST AVE	20010117	2087775236	ezequielxromero@gmail.com
10000049	841023438	5f4dcc3b5aa765d61d8327deb882cf99	Terry Lenette	9 Longbranch Dr.	19921111	2086546514	loscar@aol.com
10000048	121706986	5f4dcc3b5aa765d61d8327deb882cf99	Gayla Desiri	Pensacola, FL 32503	19750506	2082845248	gknauss@att.net
10000047	873603718	5f4dcc3b5aa765d61d8327deb882cf99	Roze Sheba	East Haven, CT 06512	19830212	2085944363	afifi@me.com
10000046	202269449	5f4dcc3b5aa765d61d8327deb882cf99	Cherish Anett	370 North Del Monte Dr.	19920612	2085466569	boser@sbcglobal.net
10000045	202228064	5f4dcc3b5aa765d61d8327deb882cf99	Shelley Mechelle	26 Bay Dr.	19851124	2087852106	sarildder@live.com

Figure 2: List of example patients

2.2 Insurance Plans

The details of a specific insurance plan are spread out over the Coverages, IncludedProducts, InsCompany, and InsPlans tables. The InsPlans table contains basic information about insurance plans, such as the company offering them and the annual premium. Coverages contains a list of service codes and what plans cover them. InsCompany contains information about insurance companies, such as their company ID, contact information and authentication information. InsCompany lists information about an insurance company, such as their contact information and authentication information. IncludedProducts lists what medical products are covered by insurance plans.

2.3 Medical Services and Products

There are several tables that contain information about medical services and medical products as such. The MedicalProducts table contains the names of medications, their manufacturer, and their product ID. The ProductCosts table shows the stocks of each medication at a pharmacy as well as the price. The Services table contains different kinds of services, such as Pediatrics and Radiology, and their cost.

2.4 Service Providers

Doctor information is in the ServiceProviders table. It contains their contact information, ID, institution, and specialty. When a doctor has an appointment with a patient, that record in ServiceRecords has the doctor's ID. If the doctor wants to prescribe a drug after that appointment, a record is made in ReferralProducts with

the product ID of the drug and the ID of the appointment in ServiceRecords. If the doctor wants to make a diagnosis a record is made in the Diagnosis table along with the ID of the appointment in ServiceRecords. If an appointment for a particular service required the purchase of certain products (such as purchasing gauze for a surgery), those products are related to that service by linking service code to product ID in the ServiceProducts table. Finally, if a patient wants to buy a medication over the counter, a record is made in the ProductTransactions table with the date, product ID, patient ID, and transaction ID.

3 FUNCTIONALITIES

The web application has four interfaces for patients, service providers, pharmacies, and insurance companies, respectively. The patients interface allows patients to schedule appointments, see diagnoses, order drugs, enroll in or change an insurance plan, and view bills. The service providers interface allows service providers to send emails to patients, issue diagnoses, prescribe drugs, and view billing information. The pharmacies interface allows pharmacies to view their stock of drugs, view what customers have bought, and view billing information. The insurance company interface allows insurance company to view information about their customers and export billing information.

3.1 Patient Interface

The patient interface allows the user to see every doctor covered by their insurance plan, view available appointment times for that doctor, and make an appointment for a particular service. When one patient makes an appointment with a particular doctor for a particular time, no other patient can make an appointment with that doctor for the same time. Patients can also view information about several different drugs and order them. Patients can view the details of insurance plans, such as annual premium and deductible, max coverage, covered services, and product coverage. The patient can then enroll in one or switch insurance plans. Patients can view a history of their medical treatment, including name of the service provider, what facility they were seen at, the date and cost of service, and any products used or prescribed relating to this service. Patients can view a diagnosis if their doctor has submitted one. Patients can also export this information to a PDF document. Patients can also view their billing information, which contains a summary of their medical history with the costs associated with services, products, prescriptions, and drugs, as well as an explanation of how a deductible has applied to these costs. A summary of costs follows, with annual deductible, premium, employer contribution, and bill due date shown.

3.2 Service Provider Interface

The service providers interface allows the doctor to see a list of their appointments. They may issue a diagnosis for past appointments. They can also send an email message to a patient they have seen. The service provider can view a history of past appointments, including the type of service rendered, the service code, the patient and patient ID, the date, and the cost charged. The service provider can also view the total payments received from patients and insurance companies. The service provider can export all this information to a PDF document. The service provider can issue a prescription to a patient they have seen in the past.

3.3 Pharmacy Interface

The pharmacy inventory allows the user to see a list of stocked drugs as well as the product ID, manufacturer, unit cost, and number of units stocked. They can also see a list of what drugs have been prescribed to patients, including name of patient, date of sale, and cost. They can also view total revenue from drug sales as well as export this information to a PDF document. They can also send an email to a patient who has purchased a drug previously.

3.4 Insurance Company Interface

The interface for an insurance company allows the user to view information about customers and billing. The user can view information about all customers sorted by which insurance plan they are covered under. The user can view customer contact information, as well as patient ID and social security number. The insurance company can view a record of all patient medical history and medical bills with insurance applied. The company can view the total amount of insurance payouts for each customer as well as total insurance payouts for each insurance plan. The company can also view the total amount of insurance payouts for all insurance plans the company offers. All this information can be exported in a PDF document.

Pediatric Heart Transplant

[Service ID]	10000366
[Date]	12/18/2020
[Service Cost]	$2698
[Original Cost]	$2698
[Current Deductible]	$4227
[Total Cost]	**$2698**
[Due Date]	01/18/2021

Pediatric Pulmonology

[Service ID]	10000367
[Date]	12/18/2020
[Service Cost]	$1843
[Original Cost]	$1843
[Current Deductible]	$1529
[Total Cost]	**$1529**
[Due Date]	01/18/2021

Lung Transplant

[Service ID]	10000368
[Date]	12/18/2020
[Service Cost]	$2542
[Products Used]	
Pandel	
[Cost]	$746
Doxycycline	
[Cost]	$720
[Original Cost]	$4008
[Current Deductible]	$0
[Total Cost]	**$0**
[Due Date]	01/18/2021

Figure 3: Billing statement example

4 DESIGN CHOICES AND PLATFORM

This web app was designed with PHP serving webpages with data from a MySQL database. These technologies are straightforward and have a wealth of online documentation and debugging information. The code for the web frontend consisted of 28 PHP files and 2 CSS files in one directory. The small scale of this project meant that a flat hierarchy of source files would not become a hindrance.

4.1 PHP Interface

Most PHP files include the "server.php" file which sets up a database connection. The "index.php" file returns different elements depending on whether the user is a patient, pharmacy, service provider, or insurance company. Many processes such as setting up an appointment take place over multiple pages and multiple PHP files which communicate with each other through passing along data in POST requests.

4.2 Python Database Generator

To populate our database with random, plausible data, we created a python script to generate data for each table. This dummy data is formatted in such a way, that all that was required was a simple copy and paste into the MySQL environment. Not only did this populate our tables, it was able to create new data each time. This and the fact that each column in each table was customizable truly helped us when configuring the frontend of the website.

```
Patients = open("xout_patients.txt", "w")
InsPlans = open("xout_insplans.txt", "w")
InsCompany = open("xout_inscompany.txt", "w")
Coverages = open("xout_coverage.txt", "w")
Services = open("xout_services.txt", "w")
ServiceCosts = open("xout_servicecosts.txt", "w")
Membership = open("xout_membership.txt", "w")
ProductTransactions = open("xout_producttransactions.txt", "w")
IncludedService = open("xout_includedservice.txt", "w")
IncludedProducts = open("xout_includedproducts.txt", "w")
ServiceProviders = open("xout_serviceproviders.txt", "w")
ServiceRecords = open("xout_servicerecords.txt", "w")
ServiceProducts = open("xout_serviceproducts.txt", "w")
MedicalProducts = open("xout_medicalproducts.txt", "w")
ProductCosts = open("xout_productcosts.txt", "w")
Pharmacies = open("xout_pharmacies.txt", "w")
```

Figure 4: Database exports

5 CURRENT LIMITATIONS

5.1 Insurance Plan

One limitation is that when patients switch insurance plans, their billing information assumes that they have always been insured by the new insurer. To fix this, we might implement an SQL table that records the relationship between patients and insurance plans. For example, we could name this table InsHistory. In this table, there would be four columns - the first for the primary key history ID, the second for the patient ID, the next for the insurance plan ID, and the last for the date. Whenever you would choose an insurance plan,

this would be queried with an insert statement with the given patients ID, their chosen insurance plan ID, and the date. The history ID would simply act as the unique identifier of each insurance plan selection, even if the patient switches to a plan they've been on before. After implementing this, we would apply the insurance plan on each service record based on the date. If the service was performed after a certain date a patient received insurance A but before a patient received insurance B, the service would apply the properties of insurance A.

5.2 Cancelling/Refunding Purchases

Once a patient purchases a medical product, or schedules an appointment with a service provider, they can no longer cancel it. This is a total oversight. Of course, in this project, no money is actually being spent. Instead, patients are given a billing statement, which has its own due dates. In this case, it is up to the patient to deliver the money, either through check or cash to the actual hospital, all hypothetically of course; however, even with this in mind, there is no structure yet in place that allows the patients to cancel their orders or appointments. To fix this, it would seem all we would have to implement is a DELETE FROM query, in which we delete the existing service record or transaction record, given that the service hasn't occurred yet and that the product from the transaction hasn't been received yet.

6 FUTURE IMPROVEMENTS

6.1 Pharmaceuticals

Currently, the patient is allowed to purchase all pharmaceuticals off the market, despite not having a prescription with said drug; instead the prescription is included in their service record, in which any insurance plan's deductible is able to apply, effectively covering the cost of the product. A much better way to handle this would simple be to allow the patient to purchase drugs which they are prescribed ONLY. This would mean prescriptions would not be added as a service record, but as a transaction record.

6.2 Pharmacies and Pharmaceuticals

When purchasing a drug, a patient is given a set of all possible drugs sold. These pharmaceuticals, however, are unique to each pharmacy, meaning pharmacies carry their own unique inventory of products. This, of course, does not emulate real systems within and

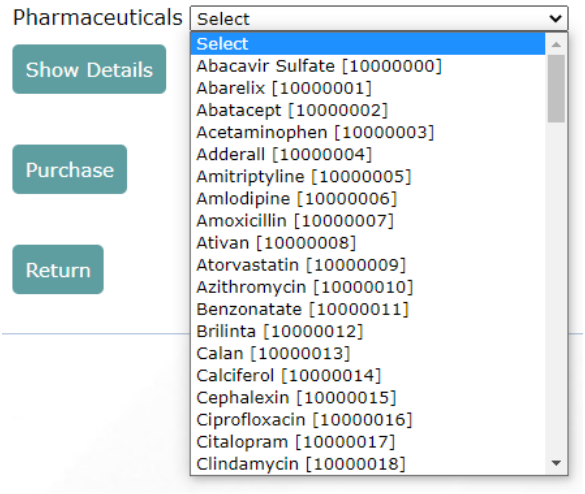

Figure 5: Database exports

between pharmacies. It is more than likely that multiple pharmacies will carry the same product, sometimes under another synonymous name. This would also be a function. This could easily be implemented through the use of two more tables. Currently, the only table responsible for this setback is ProductsCosts. This is not to say that this table is flawed; instead, it is incomplete. We can improve this method by creating two new tables named ProductType and ProductSynonyms. ProductType would be responsible for determining the specific product for each pharmacy. This would allow an overlap of products between pharmacies. Next, ProductSynonyms, given a product ID, would simply list a few other names for the same drug. After implementing these two, it would be very simple to purchase similar medical products from different pharmacies.

6.3 Pharmaceutical Price Changes

At this time, the pharmacies' interface simply presents their product stock, shows their patient drug usage, and allows them to contact their patients. However, a great addition to this would be to allow the manufacturers to change their prices. This might entail an entirely new interface for manufacturers, but maybe not. The manufacturers would be assumed to have their own websites which communicate with ours. With this, these manufactures would have access to the product costs and possible the product stock.

7 CONCLUSION

While testing and presenting this application we found that it possesses most of the useful features that would

be necessary for a health insurance portal. A new user can join the service and arrange for health services from doctors and pharmacies. Patients, doctors, pharmacies, and insurance companies can all view billing information and export it to PDF documents. Although it would be wise to add some restrictions and features to prescription and drug ordering, this project has most of the features necessary to facilitate healthcare.

8 SOURCE CODE

The source code for this project is provided at https://github.com/AdamMOdell/myhealthportal.

REFERENCES

[1] 2020. SNOMED CT. https://en.wikipedia.org/wiki/SNOMED_CT. (2020).

[2] Ly Zhang Renee Garett, Jason Chiu and Sean D. Young. 2017. A Literature Review: Website Design and User Engagement. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4974011/. (2017).

[3] Ezequiel Romero. 2020. Database Generator. https://repl.it/@EzequielRomero/Database-Generator#main.py. (2020).