Table of Contents

[Document Format 4](#_Toc336000344)

[Common Features and Requirements 4](#_Toc336000345)

[Certain data display and print format 4](#_Toc336000346)

[Feature Requirements for Different Versions 5](#_Toc336000347)

[Version 1.0 todos 5](#_Toc336000348)

[Version 1.1 7](#_Toc336000349)

[Claim Inquiry 9](#_Toc336000350)

[Main Claim Logic 9](#_Toc336000351)

[Patient UI 10](#_Toc336000352)

[Insurance UI 10](#_Toc336000353)

[Service UI 11](#_Toc336000354)

[Anesthesia Charge Calculation 11](#_Toc336000355)

[Claims Follow-up UI 13](#_Toc336000356)

[Today Action Items (TAI) 18](#_Toc336000357)

[Bill Generation 21](#_Toc336000358)

[EDI 22](#_Toc336000359)

[Paper Mail 22](#_Toc336000360)

[Fax 22](#_Toc336000361)

[Details of how delayed billing works 23](#_Toc336000362)

[Implementation 23](#_Toc336000363)

[Bill for Secondary Insurance 24](#_Toc336000364)

[Reports 24](#_Toc336000365)

[For 0.9: 24](#_Toc336000366)

[For version 1.0 25](#_Toc336000367)

[Statements 27](#_Toc336000368)

[Patent Ledger 27](#_Toc336000369)

[Statement UI 29](#_Toc336000370)

[Pending Statement 29](#_Toc336000371)

[Statement Generation 31](#_Toc336000372)

[Statement Templates 32](#_Toc336000373)

[Reminder statement 32](#_Toc336000374)

[Patient pre-pay before procedure 32](#_Toc336000375)

[Documents Management 33](#_Toc336000376)

[Data Administration and System Security 36](#_Toc336000377)

[Web Folder Security 36](#_Toc336000378)

[SSL 36](#_Toc336000379)

[Logon Security Administration 37](#_Toc336000380)

[User Rights Management 37](#_Toc336000381)

[Options Table and INI File 37](#_Toc336000382)

[Ini File 37](#_Toc336000383)

[Options Table 38](#_Toc336000384)

[Workflows Scenarios 39](#_Toc336000385)

[Scan documents into PDF files on Windows 39](#_Toc336000386)

[Entering new service/claims for an existing patient 39](#_Toc336000387)

[Enter new patient/service/claim 40](#_Toc336000388)

[Receive a EOB and payment by mail 40](#_Toc336000389)

[Received additional payments from insurance 40](#_Toc336000390)

[Received a EOB, but not payment 40](#_Toc336000391)

[Received a payment by direct deposit 41](#_Toc336000392)

[Received a fax offer for a claim and payment agreement 41](#_Toc336000393)

[Received a EOB with denial 41](#_Toc336000394)

[Received a request from payer for W9 41](#_Toc336000395)

[Receive a request from insurance that a change is needed in the claim 41](#_Toc336000396)

[TAI processing 41](#_Toc336000397)

[Document Templates 41](#_Toc336000398)

[Fax Cover Sheet 41](#_Toc336000399)

[Type I Statement 42](#_Toc336000400)

[Type II Statement 42](#_Toc336000401)

[Type III Statement 42](#_Toc336000402)

[Provider Report 42](#_Toc336000403)

[DB Related Details and Changes 42](#_Toc336000404)

[DB field length 42](#_Toc336000405)

[Accendo Account for DB connection 43](#_Toc336000406)

[DB field values 43](#_Toc336000407)

[Create some DB Views to help with 43](#_Toc336000408)

[DB fields with default values 44](#_Toc336000409)

[Pending DB fields changes 44](#_Toc336000410)

[DB structure changes need to be made on production 44](#_Toc336000411)

[Making DB schema changes without impacting data 44](#_Toc336000412)

[DB Schema Changes 44](#_Toc336000413)

[Actions Triggered by DB Events 46](#_Toc336000414)

[Productionization 47](#_Toc336000415)

[Main Page UI Layout 47](#_Toc336000416)

[Claim Management 48](#_Toc336000417)

[Data Management 48](#_Toc336000418)

[System Management 49](#_Toc336000419)

[Main Claim UI Data Validation and Click Order 49](#_Toc336000420)

[Nullness enforcement 49](#_Toc336000421)

[Common data type restriction 49](#_Toc336000422)

[Patient 49](#_Toc336000423)

[Insurance 50](#_Toc336000424)

[Service 51](#_Toc336000425)

[Claim Follow-up 52](#_Toc336000426)

[Source Code Documentation 53](#_Toc336000427)

[Automated Regression Testing 54](#_Toc336000428)

[Source Code Management 54](#_Toc336000429)

[NetBeans Mercurial Setup and Configuration 54](#_Toc336000430)

[Commonly used Mercurial commands in NetBeans 56](#_Toc336000431)

[Workflow 56](#_Toc336000432)

[Mercurial and Bitbucket References 57](#_Toc336000433)

[Test Plan 57](#_Toc336000434)

[Open issues need solution 57](#_Toc336000435)

[Scanning Integration 58](#_Toc336000436)

[Document acquisition UI design and one-off documents 58](#_Toc336000437)

[References 60](#_Toc336000438)

[Openemr 60](#_Toc336000439)

[Zipinfo.com 60](#_Toc336000440)

[mySQL Log 61](#_Toc336000441)

# Document Format

For each section, “Features and Logic” describes the design of how this UI page should work. “Implementation” describes the items that are not working as designed and need to be fixed. In “implementation”, the red entries are the ones with high priority that need to be fixed ASAP.

“Common Features and Requirements” section below describes features/requirements that apply throughout the system.

Unless specifically noted, all features and logic described in the document are for version 0.9.

# Common Features and Requirements

All date needs to be in the format of MM/DD/YYYY when displayed or entered. All dates fields need to have a widget that allows easy date entry. The widget also needs to allow manual date entry if the biller chooses to do so. It is Ok if we do not have the datepicker widget and require manual date entry in version 0.9. In the case of manual entry, we need to let the user know the format we are expecting, MM/DD/YYYY. When dates are part of a file name that we create, we will use the format of YYYYMMDD to help with sorting in the OS file system.

We need our UI to work across all three major browsers, IE, Chrome and Firefox, IE being the primary browser we need to support.

### Certain data display and print format

Phone/Fax numbers: (###)###-####  
SSN: ###-##-####  
Date: MM/DD/YYYY  
Time: ##:##  
5 digit zip code: #####  
5+4 digit zip code: #####-####

When the above data are stored, we only store the digits in the DB, not the “-“s.

The UI needs to have something like <http://javascript.internet.com/forms/format-phone-number.html> that enforce the format of (###)###-####.

For all of the above data, we need to provide a format cue to the user so they know what is expected. An example is to display (###)###-#### for phone/fax number. This is not critical, just a very nice to have.

For dropdown selection in many parts of the UI, we need that the items made up the dropdown list sorted before displaying to the users. For dropdown lists that are people, patient and doctor etc, names, we need the sort to be on last name and display as Last\_name, First\_name, i.e. Chen, Lin, instead of Lin Chen.

We also require that we can designate any item on a dropdown list as the default value that is displayed so the user can just skip the selection to take the default value. For dropdown lists that do not have a default value, the word “Select” need to be displayed so the user can make a selection from the list. The default value, if there is one, will be defined in Options.

All time related entries, we only need hours and minutes, not seconds.

All default values need to be managed by Options table, not hardcoded into the system.

Some of dropdown list are quite long and some of them will get longer in the future. In 2.0, we need a way to better manage the list.

We need to make the Notes UI does not take that much space, unless its content requires it.

We need our code to work on both Windows and Linux based apache servers. In order to do that, all references to file path need to use ‘/’ that works for both Linux and Windows file systems.

In 1.0, we need to have input validation to make sure the system is protected from bad data.

For all lists that display claims, we need to make sure no claim takes up more than one line.

We have the following three types of fields. All counts include space, comma in the string

1. Fixed length: All dates are fixed at 10 chars
2. Fixed max length:
   1. MRN: 8
   2. Currencies: 8 (#####.##)
   3. Provider and Facility: 8
   4. Issues and Claim Status: 30
   5. Last(Future): 3 (###)
3. Variable length that we need to chop off at the end
   1. Names: 15 + 2 for “ +”
   2. Insurance: 15

# Feature Requirements for Different Versions

This is a list of features we need in each versions of the system. While I do not expect major changes, some minor changes from this list are inevitable as we go through the work. Minority of the features I listed under a particular version are flexible, I have noted them so we know if we can move them to a later version if we have to.

## Version 1.0 todos

1. Save only what has been changed and rebill
2. Statement
   1. Trigger
   2. Statement Generation
   3. PDF statement generation
   4. Update Ledger to add statements
   5. Update data management to allow patient and service delete
3. Reports
   1. In summary report, do not count the claim unless it is billed
   2. download
4. Complete data validation
   1. All Main Claim Window date and time
   2. 01/29/1943 can not be entered into DOB
5. Claim Follow up page
   1. Add Open\_secondary\_insurance\_not\_billed and Open\_secondary\_insurance\_billed
   2. Calculate balance due by adding Amount Paid and adjustments together. Subtract from Total Charge
   3. When Balance due is 0, set Claim status to closed-payment as expected this is version 1.1
   4. Prompt when claim status is changed to open\_secondary\_insurance\_not\_billed and there is no EOB.pdf file or Another plan is not checked
   5. Remove Claim Status
      1. Inactive\_SelfPay. We also need to not do bill generation for insurance company of SelfPay
      2. Close\_payment below expected due to payer allowance
   6. Under EOB, add penalty and other discount. Subtract these when calculating Received amount
   7. PIP fee calculation based on state. This is version 1.1
   8. add Provider, Insurance Company name and insurance ID and total charge on the middle display
   9. Change the payment vs expected amount check to amount received
6. Bill Generation
   1. Fax
   2. Secondary insurance billing
      1. New Claim Status open\_secondary not billed and open\_secondary billed. Bill Generation list include all claims that has status of open\_secondary
      2. Display on Bill Generation list as Secondary Bill
      3. Regardless of the insurance, secondary bill is always Print CMS1500 for Mailing. When print, always print first bill, EOB.pdf and anesthesia record.
   3. Add link to Name and sort by all headers
   4. Add button to uncheck all
   5. Add to button selected/total (i.e. 20/34)
   6. Print all data fields EDI
   7. Download
7. Data Management, record deletion. I have a DB that has a lot of entries in insured table, but the tool only shows 2 of them to allow me to delete.
8. Complete code review and documentation
   1. Why the only thing calling day\_diff() is billaction
   2. Add comments to all functions definition, using Javadoc command format, extract using Doxygen
   3. Extract any logic that’s used by more than one purpose into helper.php.
9. Automated function tests
10. Makes options per provider, currently is per billing company
11. Additional work related to the above changes
    1. Make patient/service deletions to also delete the statement table record for the patients/service
12. TAI be able to sort by all headers
13. QA and bug fixes for all of the above.
14. Change insurance cause changes to claim status
15. How the class definition of DB fields is synced with actual DB fields. If there is no need to manually keep the class definition in sync with DB, we should remove all DB field definitions in the class.

## Version 1.1

#### Suppport for patient payment in installments, together with entries for biller adjustments

Add a button to Patient payments and biller adjustments that can open a popup. We need the popup to allow many entries of payments from patients who are paying by installments

#### Fill in Insurance Type using Insurance

Add custom fields on follow up

#### Make charge calculation more flexible

We need to support

1. Calculation based upon CPT code that are not in the anesthesia crosswalk. This allows the system to bill for non-anesthesia codes
2. Allow biller to enter anesthesia code directly without entering CPT code.

Allow a payment to be entered before the procedure is performed

Makes it so that a user can focus on working with a particular provider

1. Be able to default a biller ID to work with a particular provider.
   1. For Claim Inquiry, automatically select the default provider
   2. For TAI, automatically sort by provider
   3. For Bill generation, only select the bills for the provider
   4. For Statement, only select the statements for the provider
2. Add a right that is read only and limited to a particular provider.

#### Makes default modifier depend on CPT code.

Via options, if CPT code has an anesthesia crosswalk, we default to AA. If the CPT code does not have an anesthesia crosswalk, we default to BB. If the CPT code is one of the following, we default to CC etc.

#### Reminder statement Make Insurance, CPT, crosswalk per biliing company

#### Automated ERA posting with ANSI 835 file

Security and performance review

#### Full Fax functionality

Fax functions beyond just Bill Generation

#### Medicare and Medicaid billing

#### Pain management billing.

1. Scheduling
2. Pre-cert
3. Chart

#### Complete AR

1. Billing Company to Provider

Monthly/quarterly fee model based up a percentage of collection.

1. Billing Company to Insurance

Nothing in 1.0

1. Billing Company to Patient

On demand statements currently in 1

#### Misc Generate MRN with provider in it, so it is 00002934HM Check to confirm the same rendering provider’s time does not overlap with a defined time gap Online help on each UI page. Option to use 12 hour time instead of 24 hours. Limit size of document upload claim record and document archiving by age. tool to recalculate charge based on new rate for all delayed billing claims Fill in Insurance Type using Insurance Add 4 custom fields

The Claim Follow-up check with Received amount should be changed to Amount Paid. The check should be against total payments.  
If Claim Status change to Closed\_\* fails due to our prompt, we should not fill in the Date Closed.   
show numbers on the claim submit button.

# Claim Inquiry

Search UI needs to be on the same landing page after logging in below TAI.

The search UI allows the biller to enter any combination of a list of date fields to conduct claims search. If the provider is selected, Rendering Provider, Referring Provider and Facility will be limited to only those associated with the selected Provider.

If only Start Date is selected, we will only search services that that date.

The search returns a list of matching claims. For each claim in the list, the system offers the option to view the claim, which can also be updated. From the Claim Search Result list, the biller also has the option to enter a new claim for an existing patient. At the end of the list, biller can also enter a new claim for a new patient. The system automatically starts a new patient UI if the search returns no result which is populated with the last name if it is used to start the search.

If the biller does not enter any search criteria at Claim Inquiry and click Submit, the system returns all claims for the billing company. We will most likely change this to open a new claim UI so the user can enter a new claim.

# Main Claim Logic

There are four UI pages, Patient, Insurance, Service and Claim Follow-up. The biller back move among the 4 pages and fill in the data for the claim. The biller can same the claim using Save Claim or Finish Claim, which is the same as Save Claim, but also leave the Main Claims Window and go back to Claim Inquiry.

The biller can commit the changes at either the Service page or at the Claim Follow-up page. When the claim was first created, the system needs to set the claim\_status to “Open-not billed” for all payer type except medical. For medical, the initial claim\_status should be automatically set to “Open-delayed bill generation” per Options

The biller can update an existing claim by making changes to any data item of a claim, including the MRN

When a biller goes to review/update an existing claim, the system shows the Service UI page of the claim regardless whether the biller comes to the claim via query or TAI.

If biller makes changes to an existing claim, when we save the claim, there are the following cases:

1. The change was made in the Claim Follow-up section. No further action is required
2. The change was made in the Service section. If the service in question has been billed already and is still open, we need to ask the user whether rebill is needed for this particular claim after the change.
3. The change was made in the Patient/Insurance section. We need to bring up all service/claims for the patient that has been billed already that are still open and ask the biller whether any of them need to be rebilled.

The above only applies to updating an existing claim, not during the initial creation of a new claim.

To mark a claim as ready for rebill, we need to set the claim\_status to open\_not\_billed. The bill generation logic will check this and add this bill into bill generation list. See bill generation section for details related to this.

## Patient UI

#### Features and logic

There should not be default value for any of the fields, except “Relationship to insured”, which per Options, could be default to “Self”.

Use <http://www.webservicex.net/uszip.asmx> to allow biller to enter zip code and system automatically looks up and populate city and state. There needs to be a timeout built in, so if the web service is not available, the system does not hang. If we have the code already and it is easy to implement, let’s include in 0.9. Otherwise, this is 1.0.

Account number/MRN can be either auto generated or enter by the biller. This is controlled by auto\_mrn and next\_mrn in billingcompany table. will be entered by the biller, we need to check the account number to make sure it is not a duplicate within the same provider. This is done when the biller saves the claim.

## Insurance UI

#### Features and logic

Use dropdown in the Company Name to let biller select insurance company

Payer type is tied to the EDI property. More research is needed on this.

For payers with an EDI number in the DB, the system should default to that for Claim Submission Method.

The user can not change payer information, which we will maintain using phpmyAdmin for now.

Need to always keep patient data changes synced to insured if they are the same person.

## Service UI

#### Features and logic

End Date should always defaults to Start Date.

Per Options, we can let Modifier to default to AA. We need to add more logic to default modifier in 1.1

Place of Service default to facility.POS

Diagnosis Code Pointer should be related to how many diagnosis codes are entered.

When a biller enters a claim, this should be as far as the biller has to go so we need to allow the biller to commit the claim to DB by “Save Claim”.

#### Defaults for Provider, Rendering Provider, Referring Provider and Facility

When logging in for the very first time to the system, the biller has to make selection for Provider, Rendering Provider, Referring Provider and Facility for the service when creating the first claim. The system will remember the selection from the last service entered so that if you enter several services with the same selection, the biller does not have to reselect them again.

If the biller enters a new service for an existing patient who had previous service in the system already, the system will always default to the same selection as the more recent service.

If an existing service is updated in terms of the selection, the system will consider that one-off and will not try to use the changed selection as default for new services.

### Anesthesia Charge Calculation

Once we look up the anesthesia CPT code from surgery code using the crosswalk table, we find the number of base units for the anesthesia code. The total anesthesia units is the number of base unit, plus the number of time units. Anesthesia time unit is calculated by dividing the anesthesia time minutes by 15. We add the base units to time units to get the total anesthesia units, and we multiply that by the charge rate per unit to arrive at the total charge.

An example:

Anesthesia started at 8:00 AM and ended at 8:40 AM, so total of 40 mins, divided by 15 and round up, we have total of 3 anesthesia time units. The anesthesia start and end times are on the anesthesia time sheet.

The surgery CPT code is 64993, and the corresponding anesthesia code is 01935. The crosswalk table also tells us the base unit for 01935 is 5. The surgery CPT code is in the service data sheet.

The total number of units is 5 base unit + 3 time units = 8.

If the base charge per unit is $250, the charge for this claim is $250 times 8 units = $2000.

The base charge per unit needs to be defined per provider

The option on how to round when you divide the time minutes by 15 should be defined per provider

The crosswalk table is shared by all for now.

Insurance company has the option to override the anesthesia CPT code as some insurance companies do not follow the crosswalk. We have the insurance.anesthesia\_crosswalk\_overwrite to do that. It works by defining pairs of surgerycode|anesthesiacode. The crosswalk logic needs to refer to this to make sure there is not any pair defined and if there is, this pair overwrites the crosswalk table.

Cases this is needed:

Ameriprise 64490|01992 The Anesthesia Crosswalk is 01935

64493|01992 The Anesthesia Crosswalk is 01935

The expected payment from insurance should be 80% of Total Charge unless the insurance is PI and the insurance company has a fixed rate per unit. When we implement billing for in network, the expected amount is based upon the contracted rate, in which case a fee table will be needed.

In version 1.0, there could be multiple surgery CPT codes. The biller is to enter all of them and the system needs to crosswalk each of them and use the one whose corresponding anesthesia code has the highest number of anesthesia base units.

Balance Due (CMS1500:30) = Total Charge (CMS1500:28) – Amount Paid (CMS1500:29). In our current version, Amount Paid is always 0, so the Balance is the same as Total Charge

#### Implementation

Make the derived data fields on the UI not modifiable by the user, such as Anesthesia Minutes, Anesthesia Units and Charge on the Service page as well as Total Charge on the Claim page. There should be visual cue that this data is not modifiable.

We need to do units look up and charge calculation each time crosswalk code is updated. Currently the calculation is done only when time is entered. We will also need a special entry in the crosswalk table that does not do the cross and only do the base\_unit lookup without the surgery\_code. Something like if we have the surgery\_code of 99999, the system just match the anesthesia\_code and find the base\_unit.

In 1.1, we need to allow the following in charge calculation

1. Enter anesthesia code without enter CPT code.
2. Calculate charge based on CPT code that does not have corresponding anesthesia code. The charge calculation is based upon CPT code alone, so we need a fee table on a per provider basis that maps CPT code to charge.

## Claims Follow-up UI

#### Features and logic

Date Billed and Date Last Billed are filled by the system when bill generation is executed for this claim, see Bill Generation section for details.

When claim status is set to close\_\*, system automatically fills “Date Closed” with today’s date.

If the biller enters “amount insurance issued”, then “amount insurance payment received” should be left at blank. When either of the amounts is entered, their corresponding dates should be filled with today’s date.

“sent to patient” is for the biller to select. When it is selected, the Claim Status should be changed to “Open-payment issued not received”

When claim status is set to closed\_payment\_expected, system need to prompt with a list of all claims for the same patient that with claim status of open\_delayed\_bill\_generation and ask the biller to bill them by changing their claim status to “open\_not\_billed”

“Date Closed” is set by the system to today’s date when “Claims Status” is changed to “closed\* “

“Date of Initial Offer” and “Date of Agreement” need to be automatically filled with today’s date when their corresponding amounts are entered by the biller.

Need to add two DB fields for the initial amount and initial date.

When biller enters “insurance agree payment”, the system automatically enters “Date of agreement” with today’s date

#### Claim Status Logic

We have two DB tables:

a. claimstatus that has the following fields

id:

claim\_status\_display: Displayed on all UI pages that need Claim Status

claim\_status: This is the internal string our code uses

required: whether this claim status is a required one

b. billingcompanyclaimstatus that has the following fields:

id:

billingcompany\_id: link to billingcompany table

claimstatus\_id: link to claim\_status table

The claim status values that are required will always be available to a billing company and the billing company can add others as needed.

We need our management tool to allow a billing company to manage their own claim status list. The tool needs to enforce the following:

1. The required ones cannot be deleted
2. Claim Status has to start with Open, Closed or Inactive

The values have to be sorted before they are displayed on the dropdown list.

“Claim Status” allows the following values. The value will be either selected by the system as a result of some events, or biller can pick from a dropdown.

Open\_not\_billed. Set by the system as the claim is saved into the DB by the biller.

Open\_billed. Set by the system as the bill is generated

Open\_rebilled . Set by the system as the last bill date is updated.

Open\_payment\_issued\_not\_received. Set by system when “Amount insurance Payment is issued” is filled, but “Amount insurance payment received” is not.

Open\_payment\_issued\_not\_received\_check\_requested. Set by system when type II statement is generated for the claim

Open\_payment\_agreed\_not\_received. Set manually by biller in 0.9. Set by the system in 1.0 when only initially amount and date are entered.

Open\_follow\_up\_payment\_below\_expected. Set manually by biller.

Open\_follow\_up\_pending\_appeal . Set manually by biller.

Open\_follow\_up\_pending\_appeal\_results. Set manually by biller.

Open\_follow\_up\_other. Set manually by biller

Open\_delayed\_bill\_generation. Set automatically by system when the insurance.payer\_type is “MM”

Open\_pending, set manually  
Open\_Reprocessing, set manually  
Open\_investigation\_continuing, set manually

Inactive. Set manually by biller  
Inactive\_self pay, set by system when the insurance.payer\_type is “SP” at bill creation.

Closed\_payment\_expected. Set manually by biller

Closed\_payment\_below\_expected\_payer\_allowance. Set manually by biller

Closed\_payment\_below\_expected\_patient\_responsibility. Set manually by biller

Closed\_payment\_below\_expected\_no\_precert. Set manually by biller

Closed\_appeal\_turned\_down. Set manually by biller

Closed\_denied\_fund\_exhausted. Set manually by biller

Closed\_denied\_no\_policy\_in\_force. Set manually by biller  
Closed\_denied\_no\_precert. Set manually by biller

Closed\_denied\_Independent\_Medical\_Exam. Set manually by biller

Closed\_denied\_Physician\_Advisory\_Review. Set manually by biller

Closed\_denied\_Decision\_Point\_Review. Set manually by biller

Closed\_denied\_Medical\_Director\_Review. Set manually by biller

Closed\_denied\_other. Set manually by biller

The bill generation logic needs to check for the above flag so to know to list this claims to generate the bill again. When it does, it needs to update the last bill generated date with the date of this bill generation.

#### Pre-cert and pre-payment

When start and end times are not entered, we leave Claim status as empty to indicate this procedure has not been done.

We add an entry at the top of page for precert that record pre cert. The biller can also enter a pre-payment below in patient payment area.

#### Basic in network vs out of network billing

In the case of out of network, or non par, there is no pre-agreed rate at which the payer will pay the provider per anesthesia unit. The insurance will take the charge coming from the provider and apply what’s call reasonable and customary charge to decide the amount of pay. This amount is sometime negotiated between the payer and provider on a per payment basis.

In the case of in network billing, each provider negotiates with commercial insurance to reach a in network billing rate per anesthesia unit so all rendering providers in the same provider will have the same in network rate from a given payer. This rate is negotiated one time and used for all payments. The agreed rate should be the same for each Rendering Provider in the Provider so we can have a renderingprovider.in\_network\_payers that represent all of payers that this renderingprovider is in-network. We can probably use a comma separated string, like “cigna,united healthcare, oxford”, where the name matches the insurance table (We probably should not do this, and use a link instead)

In-network or out of network is only pertaining to insurance of payer type MM

The way Charge and Payment Expected are calculated is as the follows:

1. If this bill is out of network, based upon renderingprovider.in\_network\_payers, the per unit rate used to calculate how much to charge is options.anesthesia\_billing\_rate\_for\_non\_par.
2. If this bill is out of network, and the payer type is MM, the Payment Expected is options.non\_par\_expected\_pay of Total Charge
3. If this bill is in network, the per unit rate used is options.anesthesia\_billing\_rate\_for\_par and use that to calculate Charge
4. If this bill is in network, the Payment Expected is calculated by looking up the contract rate for the payer using contractrates table

If the payer type is PI and LI, the Payment Expected is options.PIP\_rate times anesthesia units. If the payer type is WC, the Payment Expected is calculated the same as out of network.

|  |  |  |  |
| --- | --- | --- | --- |
| In Network (renderingprovider.in\_network\_payers) | | Out of Network (renderingprovider.in\_network\_payers) | |
| Charge | Payment Expected | Charge | Payment Expected |
| Options.anesthesia\_billinh\_rate\_for\_par | options.in\_network\_contract\_rates X Units | options.anesthesia\_billing\_rate\_for\_non\_par | options.non\_par\_expected\_pay X Units |

Innetworkpayers is a table that links multiple insurance to multiple renderproviders.

Add options.anesthesia\_billing\_rate\_for\_par, which is used the same way as options.anesthesia\_billing\_rate\_for\_non\_par

Contractrates tables contains information about which provider has contract rate with which insurance and what the rate is.

Based upon the data in innetwork payers, we display under Benefits on Claim Followup UI page whether this partular claim is in network or out of network. This UI display cannot be changed by the biller.

We need the Data Management UI to manage these fields.

#### ANSI 835 Processing

Loop 1000A: Payer

Loop 1000B: Provider

Loop 2100: Represent a claim. There can be more than one 2100 loops.

Loop 2110: Represent a service line under the claim, can have up to 6 2110 loops for each 2100

**Exceptions we need to account for and log:**

1. There is a payment in the Ansi 835 file that we do not find a matching claim for
2. The matching claim we found has EOB data already
3. There is no valid claim data in the file that was selected.

The UI to pick Ansi 835 files need to allow selection of multiple files. On the same UI, we need a log, similar to Bill log that uses the same format as bill log, but at the end add

Ansi 835 file name “Success”, “No matching claim”, “Matching claim has EOB data” or “No claim in file”

In the case of “No claim in file”, the only fields we have for the log entry will be Date, Time, Ansi 835 file name and “No Claim in file”

**We need to use the following to match a claim, if any one does not match, it is not a match for the claim:**

1. Provider Tax ID Loop 1000B/N1/N104 if N103 is FI, or

Provider NPI Loop 1000B/N1/N104 if N103 is XX

1. Patient MRN Loop 2100/CLP/CLP01
2. Date of Service Loop 2100/First DTM/DTM02
3. Code that we billed for in the first data set. Loop 2100/2110/SVC01/sequence 2
4. Total Charge Loop 2100/CLP/CLP03

**On Claim Followup UI, we need to populate the following items:**

Allowed Amount Sum of Loop 2110/AMT02  
Not Allowed Amount Sum of Loop 2110/CAS03  
Co-insurance Sum of Loop 2110/  
Deductable Sum of Loop 2110/  
Reduction Sum of Loop 2110/  
Other Reduction Sum of Loop 2110/  
Adjustment Reason. Loop 2100/Loop2110/CAS02, claim.EOB\_adjustment\_reason look up using <http://www.wpc-edi.com/reference/codelists/healthcare/claim-adjustment-reason-codes/> and display below the tables on the PDF file. I will make a table for this.

**We need to make a EOB.pdf and Payment.pdf files and place them in the right folder for them to show on Claim Followup UI**

We will use tables as illustrated by the “Table with MultiCells” script on [www.fpdf.org](http://www.fpdf.org) to present the ERA 835 data in a one page PDF file.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Payment Date: BPR/ BPR16 | Total Payment Amount: BPR/BPR02 | Payment ID: TRN/TRN02 | TIN: N1/N104 if N103 is FI | NPI:N1/N104 if N103 is XX |

**Loop 1000A and 1000B**

|  |  |
| --- | --- |
| Name: N1/N10202 Street Address 1: N3/N301 Street Address2 : N3/N302 City, State ZIP: N4/N401, N402, N403 | Name: N1/N10202 Street Address 1: N3/N301 Street Address2 : N3/N302 City, State ZIP: N4/N401, N402, N403 |

**Loop 2100**

|  |  |  |  |
| --- | --- | --- | --- |
| Name:NM103, NM104 | MRN: CLP01 | Payment: AMT02 |  |

**Loop 2110**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| DOS | Code | Charged | Not Allowed | Allowed | Co-insurance | Deductible | Payment | Remark |  |
| DTM02 | SVC01-2 | SVC02 | CAS03 | AMT02 of AMT01=B6 | CAS06 if CAS05 =1 | CAS09 if CAS08=2 | SVC03 | AB, CD |  |
| DTMo2 | SVC01-2 | SVC02 | CAS03 | AMT02 if AMT01=B6 |  |  | SVC03 | EF |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

AB:  
CD:  
EF:

#### Implemen tation

1. On EOB, we need the logic to calculate “allowed amount” or not “allowed amount” given the other. If total charge is $100, the biller put in $30 for “not allowed amount”, the system need to put in $70 into “allowed amount” .
2. Amount Paid should equal to total of all payment
3. On benefit, when any of the amounts are changed, we update benefit\_date\_taken to today
4. If Claim Status is changed to Open-follow-up-\*, the system automatically set Next Follow-up to one week from today
5. Need to add Adjustment Reason under EOB. We need this to take up the entire line as the second line in EOB
6. Move “EOB” and “Benefits “up to the top blank line of their sections. Change EOB to Explanation of Benefits

#### Confirmation prompts for Claim Status Change:

1. Change to close-payment as expected
   1. If “Allowed Amount”, or “Amount Paid” if “Allowed Amount” is empty, is less than “Payment Expected” , we need the Confirmation prompt
   2. If “Allowed Amount”, or “Amount paid” if “Allowed Amount” is empty, is less than “Agreed Amount” , we need the Confirmation prompt
2. Change to close-\*
   1. If there is no EOB file
   2. If “Allowed Amount” is not empty and “Amount Paid” does not equal to “Allowed Amount” minus “Deductible”, “Co-Insurance”, “Reduction” and “Other Reduction”, we need Confirmation prompt.
3. Changes to Open – Secondary Insurance not billed
   1. If there is no EOB file or if there is no Other Plan checked and under Other Plan, Insurance is selected and Policy number is entered. “

This Confirmation prompt needs to have a line stating why we prompt and a space for the biller to put in why he confirms the entry or to cancel. If the biller enters the confirmation, we allow the biller to do what he wanted, but log an entry into the Interaction log. If the biller chooses to cancel, the action that triggered the prompt is cancelled and the biller is back to the normal UI.

# Today Action Items (TAI)

#### Features and logic

These are the triggers that can get a claim into TAI. The score ranges from 1 to 9, 9 is the highest. This list also has the display that we need to show on TAI UI and what the biller needs to do to remove the claim from TAI. Only claims with claim status of open will be considered on this list. This also need to exclude insurance.payer\_type of LI unless otherwise noted.

1. The claim has a “Date of Next Follow-up” of today or before today.
   1. Score: 10
   2. Display: Follow-up date <date>

**Criteria**: followup.date\_of\_next\_followup is today or prior to today.   
**<date>**: followups.date\_of\_next\_followup  
**Removal**: The biller removes or updates date of next follow-up

1. The claim has no interaction for more than X number of days, where X is configurable per provider. Interaction is defined as a follow-up entry has been entered.
   1. Score : 6.
   2. Display: No interaction for <XX> days

**Criteria**: the most recent followups.date\_and\_time\_\* is more than <X> number of days past today. If there is no follow-ups.date\_and\_time\_\* yet, the comparison is made with date\_last\_billed. If date\_last\_billed is empty, the comparison is made with date\_billed. If date\_billed is empty, the result of the comparison is false.  
**<X>**: options.number\_of\_days\_without\_activities. Default 30  
**<XX>**: Number of days without interaction.  
**Removal**: Biller adds a more recent interaction log entry

1. The claim has no payment issued after Y number of days from date\_billed or date\_last\_billed
   1. Score 6
   2. Display: No payment for <YY> days

**Criteria**: the claim.date\_last\_billed or date\_billed, if date\_last\_billed is null, is more than <Y> number of days past today.  
**Y**: options.number\_of\_days\_no\_payment\_issued. Default 60  
**<YY>:** Number of days without payment issued  
**Removal**: biller puts in the date payment is issued.

1. The claim has payment issued, but not received after Z number of days
   1. Score 8
   2. Display: Payment issued, but not received for <ZZ> days

**Criteria**: the claim.date\_insurance\_payment\_issued is <Z> number of days past today with claim.date\_insurance\_paymen\_received null  
**<Z>:** claims.number\_of\_days\_after\_issued\_but\_not\_received. Default 5  
**<ZZ>:** Number of days after issued but not received.  
**Removal**: the biller updates the date of payment receipt by posting amount received.

1. Claims associated with litigation are alerted every W number of days to check with the attorney’s office. Use claims.date\_last\_checked\_with\_attorney
   1. Score 2
   2. Display: <WW> days without checking with attorney office

**Criteria**: If insurance.payer\_type=LI and claim.date\_last\_checked\_with\_attorney is more than <W> number of days past today. If claim.date\_last\_checked\_with\_attorney is Null, the comparison is made with Date last billed or Date Billed if last billed is empty  
**<W>:** options.number\_of\_days\_for\_litigation\_followup  
**<WW>:** actual number of days since last checked  
**Removal**: biller update claim.date\_last\_checked\_with\_attorney

**Need to add UI on Claim Follow**-up for claim.date\_last\_checked\_with\_attorney. The UI box should be next to Next Follow-up Date. Let’s call it “Date Attorney Check”

1. There is an agreed upon payment, but no payment has been received for M number of days.
   1. Score: 8
   2. Display: Payment agreed, but not received for <MM> days

**Criteria**: follow-ups. date\_negotiated\_amount\_reached is more than <M> number of days past today, assuming claim.amount\_insurance\_payment\_received is empty.  
**<M>:** options.number\_of\_days\_no\_payment\_after\_agreed. Default 15  
**<MM>:** number of days passed follow-ups.date\_negotiated\_amount\_reached  
**Removal**: update claim.amount\_insurance\_payment\_received

1. There is an initial insurance offer, but we have no agreement for N number of days
   1. Score:8
   2. Display: Payment offered, but no agreement for <NN> days

**Criteria**: followups.date\_initial\_offer is more than <N> days past today with followups.negotiated\_payment\_amount=Null  
**<N>:**options.number\_of\_days\_offered\_but\_not\_agreed. Default 2  
**<NN>:** actual number of days after offer, but without agreement  
**Removal**: biller enters Insurance agreed payment and date of agreement

**NEED to add DB field** options.number\_of\_days\_offered\_but\_not\_agreed

1. The bill is ready to bill, but no bill generation for more than O days
   1. Score:8
   2. Display: No Bill Generation for <OO> days

**Criteria:** claim.date\_creation is more than <O> days past today, assuming claim.date\_billed =Null **<O>:** number\_of\_days\_bill\_has\_not\_been\_generated. Default 2 **<OO>:** Number of days a claim is not billed **Removal:** biller enter the agreed amount

All of the above variables have their default value defined in Options table. Please see Options table section below for details

Items 5, 7 and 8 can wait until 1.0. 0.9 needs to implement the rest.

TAI should list the claims with the highest score on top. For claims that have the same score, sort by total charge amount, with the largest amount on top. If a claim meets more than one of the above triggers, it should be listed only once at the sort order of the highest score.

Only claims that have the claims status of open\* are eligibility for TAI, we should consider using a DB View that only includes claims that the claim.claim\_status are open-\* with date\_billed not null. This will exclude the claims that are either closed already or open but not yet billed.

The list of claims to display in TAI needs to show Last name, First Name and DOB of the patient. Also provider, rendering provider names, DOS, payer name and display which one of the above triggers that the claims is being displayed in TAI using the Display description. We need to be able to sort by two of the fields, Score and Payer. The default sort order is by Score, with the highest on top of the list. We need to allow the biller to click the heading and change the sort order to Payer. The ability to sort by payer is to allow the biller to discuss all claims in TAI in a single interaction.

When a claim is selected from TAI, the claim should be displayed in the main claims UI on the Service page. The biller can move back and forth through the UI pages for the claim and make changes. All features we discussed in Main Claims UI are required here.

In version 0.9, the list is generated when the biller logs in. For 1.0, we need a button to refresh the list.

We should create a new TAI table that we manage manually. The table has the following fields:

ID  
Score of the trigger  
Display in TAI for the trigger  
Can we expression our criteria using regular expressions and have TAI entirely data driven? That would be best!

We also need to add a field in Provider table to link to this table, similar to Options table.

#### Implementation

5, 7 and 8 above will wait until 1.0. We will need the rest to work in 0.9

Need a way to do exceptions for 2 and 3 on a per insurance basis, so that even though the default for 2 is 30, we need Horizon to be 7.

Making TAI data driven

Need to add DB field and UI for initial offer amount and date to support item #7

# Bill Generation

#### Features and logic

A claim is ready for bill generation if:

1. Claim\_status = “Open-not billed” and claims\_date\_billed is null. This displays “New Bill”
2. Claim\_status = “Open\_not billed” and claims\_date\_billed is not null, which means the claim is ready for rebill. This displays “Rebill”.
3. Claim\_status = “Open-delay bill generation”, but we have passed the length of delay as defined in Options. The system compares DOS with today’s date to know whether the delay has expired. This displays “Passed delay time”

The application needs to display all claims that are ready to bill. This should be all existing claims in the DB that meet any of the above conditions. The list should display Last Name, First Name, DOB, DOS, Rendering Provider, why it is ready for bill and which one of the billing methods is used, EDI, Mail or Fax.

The list should be sorted by the type of billing, EDI/Mail/Fax. We need to have a check box for each claim that allows the biller to check or uncheck so included or not included in this particular bill generation.

Once a bill is generated for a claim, the system needs to set claims.date\_billed to today’s date if this is initial bill. If this claim is a rebill, we need to set date\_last\_billed to today’s date

The system need to change the claim\_status to “Open-billed” or ”Open-rebilled”

Claims.comment\_\* goes to the shaded area above dates and times for each set of data on a printed form. We need the comments to list the start time, end time anesthesia minutes. Also anesthesia base unit and time unit. I have this on the test.pdf file as a sample, so take a look.

In 0.9, the comment only displays start, end time and minutes. The calculation on the PHP side which is repeating the calculation done by Hao Wei’s code on the front in the Service UI.

For bill generation, we need to create one file for either EDI or PDF. Both EDI and PDF should be sent to the browser so the bill can easily download them.

EDI  
When the EDI button is pressed, the system should generate the ANSI 837 file. We will manually upload the file to the Clearinghouse.

Paper Mail  
We need to generate a PDF file that include all bills as well as their corresponding Anesthesia Record. This PDF needs to be downloaded via the browser so the biller can print it out to the printer.

### Fax

We will use [ebsllc@hotmail.com](mailto:ebsllc@hotmail.com) and smtp.live.com on port 25 or 587 as the originating mailbox with password of qwerty123.

For testing, please use [ebsllc@hotmai.com](mailto:ebsllc@hotmai.com) as the mailbox to send the email to. Once we have thoroughly tested and are ready to test for real, we change above [fax\_number@myfax.com](mailto:fax_number@myfax.com), where fax\_number is insurance.fax\_number with a “1” added to the front of the number. Let’s see the insurance.fax\_number is 21223456789, then the email address to send the email to is [12123456789@myfax.com](mailto:12123456789@myfax.com). Please do not use the production address until we have done all testing, as the production fax incurs a cost.

The email should have no subject and the body of the email should be empty too, with the PDF attachment. The attachment has three parts

1. A fax cover sheet. Please see Fax Cover Sheet under Document Template
2. CMS1500
3. Anesthesia Record

### Details of how delayed billing works

If insurance.payer\_type is “MM”, the bill will be delayed generation.

When a claim is initially created

If insurance.payer\_type is “MM”

Then set claim.claim\_status to “open-delayed\_bill\_generation”

Else set claim.claim\_status to “open-not\_billed”

When creating the bill generation list

If claim.claim\_status is “open-not\_billed”

Then the claim is on the list. The claims is shown as “New Bill” in the list

If claim.claim\_status is “open-delayed\_bill\_generation” AND Encounter.start\_date\_1 is more than “options.number\_of\_days\_for\_delayed\_bill\_generation” days past today

Then the claim is on the list. The claim is shown as “Passed Delay Time” in the list

Once bill has been generated for a claim, its claim status is set to “open-billed”. The CMS1500 file should also be saved in the Document Management folder for this MSN-DOS. This is needed for EDI billing as well.

### Implementation

We need to add the column to Bill Generation list to mark the state of the bill in terms of bill generation, like “New Bill”, “Rebill” and “Passed delay time”

Need to add printing of Anesthesia Time Sheet together with CMS1500. This requires document management implemented.

Need to always generate CMS1500, regardless whether this is EDI or CMS1500 and place the file into document management folder and makes it available to the UI

Need to fully implement EDI bill generation in version 1.0.

Need to implement Fax in 1.0. This needs to generate the PDF file for fax cover sheet, CMS1500 and Anesthesia Time Sheet and send them to fax server.

Download for printing needs to work for printed bills and statements.

We also need to accommodate the additional UI data items so that they print accordingly on the bill.

Bill Generation should only generate bills for this billing company, the system currently list bills from other billing company as well and this needs to be fixed.

We need to be able to sort the Bill Search Results list by Insurance, Provider, Means and Comment

### Bill for Secondary Insurance

We add a new Claim Status of open\_secondary\_insurance\_not\_billed, which is manually set by biller. The system only allows changing to this status from the Claim Follow-up UI after checking

1. there is a EOB file and
2. “Another Plan” is checked in Insurance UI.

Bill Generation will add claim with this Claim Status to the Bill Search Results list and its “Means” will always be “Mail” and “Comment” is “Secondary bill”. Bill generation uses the information on “Another Plan” to print the bill, which is a PDF with three parts

1. The original CMS1500, with the insurance company name and address replaced by what’s selected in Another Plan.
2. EOB.pdf
3. Anesthesia Record.pdf

After the bill is generated, we change the Claim status to open\_secondary\_insurance\_billed. We also need to store 1+2+3 in one PDF file in one-off folder under claims with name SecondaryBill.pdf.

# Reports

### For 0.9:

#### Basic canned report

The report UI need to have a list of all providers and let the biller to select which provider to run the reports on. This will be manual for now and needs to make it into TAI by 1.0 to alert the biller to run the reports at the beginning of a month.

The biller can run reports at any time, but we can only report on data we have for up to the previous whole month.

Display a table that lists the last 12 whole months that we have data. Let’s see the report was executed on 8/4/2011 and we have claims for this provider started on 3/2011. A Sample report is in SampleReport.docx. Our report should be in PDF.

Note that we always report on the data for the proceeding 12 whole months in our reports.

The first table lists the number of claims in the month, the amount we billed for the month and the amount we collected corresponding to these claims for the month, regardless when the collection/payments came in.

1. Calendar month
2. The number of DOS that falls into that month, using encounter.start\_date\_1
3. The amount billed for the DOS during that month, claim.total\_charge
4. The amount collected for the claims in month that have been received so far claim.amount\_insurance\_payment\_received

The second table has the total for each month the collections that come in, regardless which claim the payment was for.

1. Calendar month
2. Collection receive during the month regardless what claims they are for claim.amount\_insurance.payment\_received

The third table has the billing fee. It is calculated and collected on a quarterly basis. The fee is calculated as the total collection for the quarter time options.provider\_invoice\_rate. I used 5% in the sample document. We only display Fee Charged when we have a full quarter. A/R feature is not going to be implemented until 2.0, so we will leave Fee Paid empty for now. We will include up to 10 previous quarters in the table.

1. Calendar quarter
2. Collections received during the quarter regardless which claim they are for claim.amount\_insurance.payment\_received
3. Leave blank for now.

The forth table is Claim Details, please see the sample file for DB fields needed for this table which need to be sorted by last name.

The title of the reports has provider.provider\_name and billingcompany.billingcompany\_name

Ad Hoc Report  
provide a DB view that can be queried against so that

### For version 1.0

We use the same UI as the current Claim Inquiry.

We list the following list of fields to output. The ones at the top are always in the report output. The groups all have a check box and by default, they are not checked. If a group is checked by a biller, every fields in the group are also added to the output.

Name (Last, First)  
MRN  
Insurance  
Facility  
DOS  
Total Charge  
Amount Paid  
Claim Status

Patient  
 DOB  
 Sex  
 Street Address  
 City  
 State  
 Zipcode  
 SSN  
 Phone Number  
 Secondary Number

Provider  
 Provider  
 Rendering Provider  
 Referring Provider  
Services  
 DOS#  
 Anesthesia Minutes#  
 CPT Code#  
 Anesthesia Crosswalk#  
 Anesthesia Units#  
 Charge#  
Dates  
 Date Bill Created  
 Date Billed  
 Date Last Billed  
 Date closed  
Payments  
 insurance Received Amount#  
 insurance Received Date#  
 insurance payment notes#  
 Patient Received Amount#   
 Patient Received Date#  
 Biller Adjustment Amount#  
 Biller Adjustment Date#  
  
We send the resulting report to outside of our system, so no password is required to see the report, and use a randomly generated URL that Billing Company can send to a provider for viewing, something like [www.host.com/reports/](http://www.host.com/reports/) hG73kHhfi8gHGdhfuJ4j7h.html. We need the result table that can be sorted on any of the columns (at a minimum, Name, DOS, Claim Status). We also need a download button so the report can be downloaded into CSV file as well as a button to generate a PDF for it, which is similar to what we have today. The spreadsheet file has the same out as the HTML display.

When the biller clicks Reports, we display a dropdown that allow the biller to pick the provider. We also have two selections, one to produce the summary report and the other to produce the spreadsheet. The summary PDF file is the same as the one we had before, but including the current partial month/quarter. The spreadsheet

#### Implementation

# Statements

## Patent Ledger

Patient ledger is the aggregation of all of the money transactions from each of the claims for this patient in chronologic order

When the Patient UI is displayed, the system constructs the ledger by listing all of the Events from the claims table. The Events include all Insurance Payments, Patient Payments and Biller Adjustments

When a claim is denied, an entry is entered with Event of Insurance Denied Geico

Needs to handle rebill

Needs to handle secondary billing

Ledger should only include claims that are either open, or closed within the past X number of days, which is defined in Options table so it can be changed on a per Provider basis.

Statements include the ledger entries related to claims that are still open.

The code that aggregates all of the Events needs to be shared between ledger and statements printing.

1. Sample Patient Ledger

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date | Description | Amount | Balance | Notes |
| 08/10/2011 | Anesthesia Procedure 01992 by Dr. Peng at Advanced Spine Surgery Center, Union, NJ | 800 | 800 |  |
| 08/12/2011 | Insurance billed Geico CPT:01992 DOS:08/10/2011 |  |  |  |
| 09/02/2011 | Insurance payment received Geico CPT:01992 DOS 08/10/2012 | -400 | 400 | Check #3478765 |
| 09/02/2011 | Biller adjustment CPT:01992 DOS 08/10/2011 | -300 | 100 | Insurance reduction |
| 09/04/2011 | Patient billed statement I |  |  |  |
| 10/01/2011 | Anesthesia Procedure 01992 at Advanced Spine Surgery Center, Union, NJ | 1000 | 1100 |  |
| 10/01/2011 | Insurance billed Geico CPT:01992 DOS:10/01/2011 |  |  |  |
| 10/06/2011 | Patient billed statement II |  |  |  |
| 11/12/2011 | Insurance payment received Geico CPT:01992 DOS 10/01/2011 | -600 | 500 | Check #3478862 |
| 11/12/2011 | Biller adjustment CPT:01992 DOS 10/01/2011 | -350 | 150 | Insurance reduction |
| 11/12/2011 | Patient payment received CPT:01992 DOS 08/10/2011 | -100 | 50 |  |
| 11/12/2011 | Claim closed CPT 01992 DOS:08/10/2011 |  |  |  |
| 11/14/2011 | Patient billed statement I | 0 | 50 |  |
| 12/01/2011 | Patient payment received CPT:01992 DOS 10/01/2011 | -50 | 0 |  |
| 12/01/2011 | Claim closed CPT:01992 DOS 10/01/2011 |  |  |  |

1. Sample Patient Ledger with Insurance payment sent to Patient

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date | Description | Amount | Balance | Notes |
| 10/28/2011 | Procedure 00630 at First New Jersey Hospital, Morristown, NJ | 2000 | 2000 |  |
| 10/30/2011 | Insurance billed Oxford CPT:0063 DOS:10/28/2011 |  |  |  |
| 01/02/2012 | Insurance paid to patient Horizon $1600 CPT:0063 DOS:10/28/2011 |  |  |  |
| 01/04/2012 | Patient billed statement I | 0 | 2000 |  |
| 01/30/2012 | Patient payment received CPT:0063 DOS:10/28/2011 | -1600 | 400 |  |
| 01/30/2012 | Biller adjustment CPT:0063 DOS:10/28/2011 | -400 | 0 | Insurance reduction |
| 01/30/2012 | Claim closed CPT:0063 DOS:10/28/2011 |  |  |  |

1. Sample Patient Ledger Example of no patient billing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date | Description | Amount | Balance | Notes |
| 12/22/2011 | Procedure 00630 at First New Jersey Hospital, Morristown, NJ | 1750 | 1750 |  |
| 12/22/2011 | Insurance billed Aetna CPT:00630 DOS:12/22/2011 |  |  |  |
| 01/02/2012 | Insurance payment received Aetna CPT:00630 DOS:12/22/2011 | -1250 | 500 |  |
| 01/02/2012 | Biller adjustment CPT:00630 DOS:12/22/2011 | -400 | 100 | Insurance reduction |
| 01/03/2012 | Claim closed CPT:00630 DOS:12/22/2011 | -100 | 0 |  |
|  |  |  |  |  |
|  |  |  |  |  |

Events that result in entries into Patient Ledger:

1. New Service is entered
2. Bill is generated to insurance, claim.date\_billed and date last billed
3. Statement is generated to patient
4. Insurance payment is received, claim.date/amount/notes\_insurance\_payment\_received\*
5. Insurance payment is sent to patient, claim.date/amount/notes\_insurance\_payment\_sent and send to patient is checked
6. Patient payment is received: claim.date/amount/notes\_patient\_payment\_received\*
7. Biller adjustment is made: claim.date/amout/notes\_biller\_adjustment\*
8. Claim is closed.

## Statement UI

On Patient UI, we add “Statements” section followed by “Patient Ledger”.

Statements has the following UI:

Statement | Date | Trigger | Claim  
Remark

Each entry of the table is made up of two lines on the UI. When the user entered a new entry, the user only provides Statement (Select, I, II, III, installment), Trigger (Select, payment sent to patient, co-insurance, deductible, Self Pay), Service (Select, DOS and CPT) and Remark. The user can not enter date, which is populated by Statement when the Statement is actually printed.

The first entry under Statement section is the blank entry to allow our biller to enter a new pending statement. The user can enter everything except date. The second entry is the pending statement for this patient at this time. The user can enter or modify the remarks.

We need a way to allow the biller to delete a pending statement. We can do so by letting the biller to change the Statement to “Select”, which currently does not work.

## Pending Statement

Pending statement is a statement entry that has been triggered but the the statement table record does not have a date, which means this statement has not been generated.

Statement I can be triggered by one of the following, per Options. They trigger the Statement I as the data is entered.

1. “Send to Patient” is checked, which sets Statement to I, per option.SI\_send\_to\_patient
2. An insurance payment is posted and EOB shows a co-insurance, which sets Statement to I, per option.SI\_co\_insurance.
3. An insurance payment is posted and EOB shows a deductible, which sets Statement to I, per options.SI\_deductible
4. Insurance is selfpay, which sets statement to I, per options.SI\_selfpay
5. The biller manually sets the Statement UI to I, II, III or installment. No need for any prefixed remark.
6. Statement Interval expired for statement I or II he claim is still not closed. This check is done when the user goes to Statement

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Trigger** | **Action** | **Option control** | **Prefixed Remark** |
| 1 | Send to patient | Trigger SI | Option.SI\_send\_to\_patient | Option.statement\_I\_1 |
| 2 | Co-insurance | Trigger SI | Option.SI\_co-insurance | Option.statement\_I\_2 |
| 3 | Deductible | Trigger SI | Options.SI-deductible | Option.statement\_I\_3 |
| 4 | Self Pay | Trigger SI | Option.selfpay | Option.statement\_I\_4 |
| 5 | Biller manual entry | Trigger what’s requested | None | none |
| 6 | Statement interval elapse | Trigger SII or SIII | Option.patient\_statement\_interval | Option.statement\_II\_(1,2,3,4) option.statement\_III\_(1,2,3,4) |

#5 will always trigger Statement. Whether #1, #2 , #3 and #4 will also do is controlled by values in options. #5 marks this patient is in an installments, which means we need to send out up to 36 monthly statements. #5 needs to remember 1,2,3,4.

Statement II is generated option.patient\_statement\_interval days after statement I is generated unless the claim is closed

Statement III is generated option.patient\_statement\_interval days after statement II is generated unless the claim is closed

Statement table has

statement\_type: null, I, II, II and or installment  
date   
trigger: T1, T2, T3, T4, T15, T25, T35, T45, T16, T26, T36, T46. T25 means the biller manually picked T2. T36 means the original trigger is T3 and now the time interval has elapsed. We only need to offer the first 4 triggers on the UI for the biller to select.  
remark  
FK to encounter.id  
next\_statement: Boolean. Yes means the next statement has been created in pending state.   
installment\_date: DATE used so the system knows installment statement has been created for the month

When any of the trigger conditions is met, system adds a new entry into the statement table. The system populates statement, trigger, encounter\_id and remark fields and we make this entry available to the patient UI under Statement section. The remark field is copied from options.statement\_(I,II,II)\_(1,2,3,4,5,6) for the biller to change, remove or append.

The biller can also manually tell the system to generate statement, in which case, the UI will let the biller to fill in statement, trigger, claim and Remark.

When statement\_type is set to installment, we need to automatically create statement on or past options.installment\_statement\_date of every month until the claim is closed.

## Statement Generation

When the biller goes to Statement, similar to Bill Generation, the biller can print the statements out for mailing. The UI should display four buttons, Statement I, Statement II, Statement III and Installment with each of the patient has a check box that by default is checked. This is very much the same as Bill Generation. After the Statement is generated, the system fill in the date field for statement table entry with today’s date. The Statement Generation list should include Name, MRN, Insurance, facility, Provider, Rendering Provider, DOS and statement, which is I, II, III or installment.

The system goes through all open claims. For each claim we find whether there is any pending statements. If so, we list it on the list. If there is a generated statement for the claim, we need to know whether T6 applies and if so, we create a pending statement for II or III.

If a payment is set to Installment , which can only be set by the biller manually, Statement Generation automatically generates a pending statement once a month on a particular date, as defined in options, option.installment\_statement\_date so that it also show up on the Statement Generation list.

Document management needs to have statements stored and makes the document buttons available on Patient UI (I,II,III and installment).

Let’s say a patient has 3 claims. C1, C2 and C3. The 6 triggers about are referred to as T1,T2,T3,T4, T15 and T16, T26, T36, T46, T56. SI, SII, SIII means Statement I, II and III.

3/2/2012: claim 1 triggers SI due to T1, read create set SDTCR(1)=SI, ,T1,C1 ,option.statement\_I\_1

3/3/2012: SI printed. SDTCR(1)= S1,3/3/2012,T1,C1, UI Remark field.

3/10/2012: C2 triggers SI due to T2, set SDTCR(2)=SI, ,T2,C2, option.statement\_I\_2

3/10/2012: biller adds a remark so SDTCR(2)=SI, , T2, C2, option.statement\_I\_2 . ”some text”

3/10/2012: SI printed. SDTCR(2)=SI, 3/10/2012, T2, C2, option.statement\_I\_2 . ”some text”

3/20/2012: C1 closed

4/10/2012: System detect that SDTCR(2)(date) is 30 past today and at SDTCR(2)(Claim) is not closed, set SDTCR(3)=SII, , T26, C2, option.statement\_II\_2 .

4/10/2012: biller adds a remark so SDTCR(3)=SII, , T26, C2, option.statement\_II\_2 . ”second notice”

4/10/2012: SII printed, sets SDTCR(3)=SII, 4/10/2012, T5, C2, option.statement\_II\_2 . ”second notice”

4/14/2012: Biller manually entered statement SDTC(4)=SI, ,T15,C3, option.statement

4/14/2012: SI printed. SDTCR(4)=SI, 4/14/2012,T5,C3, “some more text”

4/20/2012: C3 closed

5/10/2012: System detect that SDTCR(3)(date) is 30 past past today and SDTCR(3)(Claim) is not closed, set SDTCR(5)=SIII,,T6,C2, “last notice”

5/10/2012: SIII printed, set SDTC(5)=SIII, 5/10/2012, T5,C2, “last notice”

## Statement Templates

The statements will include most of the patient ledger entries and one of the options.statement\_?\_#.

Only claims that are open will be included in the patient statements. The ledger displays on patient UI will also include claims that are closed within option.number\_of\_days\_include\_in\_ledger number of days.

The ledger in the statement may not include all entries in the patient ledger. If do implement this, we will use some options fields to do this.

### Reminder statement

Unlike a regular statements that are triggered and generated after at least the first insurance payment is issued, a reminder statement is manually triggered by the biller after entering a claim. Reminder statement also does not use the regular statement template. Another critical difference is that reminder statement is per claim, not per patient as the regular statements

Implementation

1. billingcompamy.statement0 (Boolean) that says whether there is reminder statement UI
2. Add claim.statement0 (Boolean) and claim.date\_statement0 DB fields and add checkbox UI on Claim Follow-up UI page for reminder statement if billingcompany.statement0 is set to true.
3. If the biller manually checked claim.statement0 to true, Statements will list the claim for reminder statement printing. After printing, the system needs to reset claim.statement0 and populate clain.date\_statement0.

options.statement\_0\_note. This use a different template then regular statements

### Patient pre-pay before procedure

Different from all of the claim payments we have built into the system, this payment is received before the procedure is performed.

This requires that we create a claim, but it will not have a start and end time as the procedure has not been performed. We need to make the following changes

1. Makes start and end time allow NULL and remove from the validation on the backend before writing to DB.
2. Create a new Claim Status open-not performed, which is the status when start and end time not filled.
3. The claim will not be billed because it does not have the Claim Status of open-not\_billed. Patient payment can be posted to the claim
4. When the procedure is performed, the biller put in the start and end time and that makes the Claim Status change to open-not\_billed, or open-delayed bill generation.

# Documents Management

#### Features and logic

In Version 0.9, we will manually scan and name the document and place them into the storage folder hierarchy. Billing System needs to pick them up when the biller request them displayed from the UI pages

In Version 1.0, we need to have document acquisition integration so that Billing System will instruct the biller to upload the pdf document and place the document into the right place for it to be accessed from the system.

Documents that are in scope for management:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| # | UI page | Name | File Path | # of pages | Acquisition | Always available | When | DB file path field |
| 1 | Claim Follow-up | Statement I | %root%/MRN-DOS/StatementI.pdf | 1 | Generated by Statement | No | Statement is generated | Claim.file\_path\_to\_statementI |
| 2 | Claim Follow-up | Statement II | %root/MRN-DOS/StatementIIpdf | 1 | Generated by Statement | No | Statement is generated | Claim.file\_path\_to\_statementII |
| 3 | Claim Follow-up | Statement III | %root%/MRN-DOS/StatementIII.pdf | 1 | Generated by Statement | No | Statement is generated | Claim.file\_path\_to\_statementIII |
| 4 | Service | Facility Sheet | %root%/MRN-DOS/FacilitySheet.pdf | 1 | Scanned in from Service | Yes | Service is entered | Encounter.file\_path\_to\_facility\_sheet |
| 5 | Service | ID | %root%/MRN-DOS/ID.pdf | 1 | Scanned in from Service | Yes | Service is entered | Encounter.file\_path\_to\_ID |
| 6 | Service | Anesthesia Record | %root%/MRN-DOS/AnesthesiaRecord.pdf | =>1 | Scanned in from Service | Yes | Service is entered | Encounter.file\_path\_to\_amesthesia\_record |
| 7 | Claim Follow-up | CMS1500 | %root%/MRN-DOS/CMS1500.pdf | 1 | Generated by Bill Generation | Yes | Bill is generated | Claim.file\_path\_to\_CMS1500 |
| 8 | Claim Follow-up | Agreement | %root%/MRN-DOS/agreement.pdf | 1 | Scanned in from Claim Follow-up | No | When a payment is offered and agreed upon | Follows.file\_path\_to\_negotiated\_agreement |
| 9 | Claim Follow-up | EOB | %root%/MRN-DOS/EOB.pdf | =>1 | Scanned in from Claim Follow-up | Yes | Insurance makes payment | Claim.file\_path\_to\_EOB |
| 10 | Claim Follow-up | Payment | %root%/MRN-DOS/Payment.pdf | 1 | Scanned in from Claim Follow-up | Yes | Insurance makes payment | Claim.file\_path\_to\_payment |
| 11 | Provider | Report | %root%/provider\_name/Report-YYMMDD.pdf | >1 | Generated by Report |  | Report is generated | Provider.file\_path\_to\_report |
| 12 | Provider | W9 | %root%/provider\_name/W9.pdf | 1 | Scanned from Provider Admin | Yes | Provider is added to the system | Provider.file\_path\_to\_W9 |
| 13 | Rendering Provider | License | %root%/rendering\_provider\_last\_name-first\_name/license.pdf | 1 | Scanned in Rendering Provider Admin | No | Rendering Provider is added to the system | Renderingprovider.file\_path\_to\_medical\_license |
| 14 | Insurance | Insurance |  |  |  |  | As needed | file\_path\_to\_insurance\_info |
| 15 | All UI Pages | One-off | Same folder as the UI page that acquires it | =>1 | Scanned in from any UI page | no | As needed | TBD |

Documents 4,5 and 6 are scanned in as one PDF file from the Service UI and the system needs to split them into the three separate files. The order is 4,5 and 6 in the combined PDF file provided from the biller.

Documents 8, 9 and 10 are scanned in individually from the Claim Follow-up UI

Documents 12 and 13 are scanned in individually from the Provider and Rendering Provider Administration UI

Document 14 is only available on a small percentage of patients.

Document 15 is for one-off documents associated with any of the UI pages. We need the document acquisition button on each UI page to be able to either acquire a fixed document, or a one-off document. The difference is the fixed document is known to the system and the system knows how to name the document and how to label the button to access them. On the other hand, the system does not know how to name a one-off document, so we have to ask the biller for its name which is used by the system to label the button and name the file. We can restrict the length of the one-off documents name to 10 chars so that we do not have a access button that has label that’s too long.

For the three document split function on Service UI, we need to make it so that our system can adapt to other facilities that might have a different set of documents. We will use DB to contain the page name and allow page changes without code change.

Facility.service\_doc\_first\_page VARCHAR(20)  
Facility.service\_doc\_second\_page VARCHAR(20)  
Facility.service\_doc\_third\_page VARCHAR(20)  
Facility.service\_doc\_forth\_page VARCHAR(20)

The code will loop through these fields and the last fields with a value is the last page, which could be more than one pages.

#### Main/Management UI Document access

We need to support all fixed documents and up to 3 one-off documents. This include Provider, Rendering Provider, Facility, Insurance, Billing Company.

#### Documents storage folder structure

The folder structure:

Sysdoc/cms1500: printing batch files for CMS1500 PDF files. These are are also downloaded when you generate them

Sysdoc/edi: batched edi files for upload to your clearinghouse  
  
sysdoc/reports: reports

Sysdoc/document

Sysdoc/document/claim for Claim Service and Claim Follow-up UI related documents  
Sysdoc/document/encounter for Service UI related documents  
sysdoc/document/insured for Insured UI related documents  
sysdoc/document/patient for Patient UI related documents

These are organized into folders named after claim.id|encounter.id|insured.id|patient.id. Under each folder, fixed documents are directly in the folder, off-off documents are under one-off folder

Not yet implemented:

sysdoc/document/facility for Facility UI related documents   
sysdoc/document/provider for provider UI related documents  
sysdoc/document/renderingprovider for Rendering Provider UI related documents  
sysdoc/document/referringprovider for Referring Provider UI related documents  
sysdoc/document/billingcompany for Billing Company UI related documents

Document acquisition

On UI pages, “document Input” button opens up a popup menu to prompt the biller to upload the document.

Use the traditional FilePicker from browser to locate the file for the particular document and upload.

The UI page will take the uploaded documents and put them into the right document management folder and gives it the correct file name for the corresponding access buttons to find them.

For one-off files associated with any of the above entities, we need the code to ask for the name of the document which is what the system uses to name the file as it is being stored in the document folder as well as to label the button for its access on the UI page.

For the Dateof type files, the UI page just need to display one link to the most current Date of file, the previous ones do not need to be displayed from the UI. The Date of format should be YYYYMMDD to help with file system sorting.

For each of the UI pages, we need to have a button labeled, Document Input. Depend on the UI page this button is located, it asks for the fixed documents and one-off document. The fixed document will simply renamed and placed in the right folder with the right name. The UI needs to make the corresponding button visible so the user can press it to have the document opened. For one-off document, the Document Acquisition UI will also need a name for the document, which is used both as the button label as well as the file name.

#### Implementation

# Data Administration and System Security

See Data Management UI section for UI requiements

## Web Folder Security

We need to maintain security on three folders

1. The main billingsystem folder. We do not allow browsing into this folder and no one can access any of the pages directly, other than public/index.php
2. The document folder. We do not allow browsing into this folder. No one can access any of the files directly, other than files in document/reports

## SSL

To enable SSL to PHP, add extension=php\_openssl.dll into php.ini file.

1. Allows our system to talk to gmail using SSL
2. Limit access to HTTPS

## Logon Security Administration

UI to add, remove and update a user and assign rights

We need logoff to work

Logon timeout so an inactive user session has to re-login after X mins. X can be stored in the application.INI file

## User Rights Management

These are users rights that can be assigned to any user, a biller account could be just a regular biller, or it could also be given the right of a Billing Company Admin.

#### Regular user/biller

This is a regular biller ID. The user can perform everything under Claim Management on the main UI page, but nothing else.

#### Billing Company admin user

This is the admin for the billing company. This user, besides able to do everything the regular bill can do, can also perform t the functions under Data Management. The data that this user can manage is limited to data related to this billing company.

This user has the right to management Patient, Insurance and Provider AR in version 2.0.

#### System wide admin

This right allows the user to perform all function under System Management. This will also have the right to management Billing Company AR in version 2.0.

# Options Table and INI File

Option table are per provider

Ini file are for system wide parameters at application\configs\application.ini

## Ini File

#### %Docroot%

Root folder of the Document Management folder

#### Fax Email Address

The email address to send fax email to, myfax.com  
The mailbox to send the email from and its password

#### Logon on Session Timeout

Set to 30 mins for now

## Options Table

This is per Provider which is linked to from the Provider table. It has the following settings.

anesthesia\_unit\_rounding. This controls how we round anesthesia unit, in 0.9 we only need to support always round up. In 1.0 options are:

Always round up, DB default  
 Round to nearest  
 Always round down  
 Do not round

anesthesia\_billing\_rate\_for\_non\_par This is the default billing rate per anesthesia unit, currently DB default to $250 per unit.

PIP\_rate, this is the PIP rate in NJ it is $65.72/anesthesia unit. This is not involved in the amount billed, only in the calculation of amount expected. No DB default

patient\_statement\_interval DB default to 30, not needed for now.

auto\_populate\_diagnosis\_pointer DB default to yes to populate diagnosis pointer to all diagnosis codes. When it is set to no, the system leaves diagnosis pointer blank for the bill to fill manually.

signature\_on\_file\_for\_signatures DB default to yes and we print “Signature on file” for all signatures on the claim

use\_DOS\_for\_all\_dates DB default to yes to use DOS for all date on the claim. When set to “no”, we leave all dates on CMS1500 empty for biller to fill in manually.

yes\_for\_assignment\_of\_benefits DB default to yes. If not, we leaves it for the biller to select

default\_end\_date\_to\_start\_date DB default to yes, which means always make the end date for DOS the same as start date. If not set to yes, we leave end date empty for biller to enter.

default\_patient\_relationship\_to\_insured DB default to “self”. When not set, display “Select” for biller to select from the dropdown

default modifier DB default to “AA”. When not set, display “Select” for biller to select from the dropdown

default\_facility default facility on the dropdown list. When not set, display “Select” for biller to select from the dropdown. No DB default

default\_provider: Not used. Use billingcompay.default\_provider instead

default\_rendering\_provider default rendering provider on the dropdown list. When not set, display “Select” for biller to select from the dropdown. No DB default

provider\_invoie\_rate currently set to 5%/0.05, This varies per provider. No DB default

number\_of\_days\_without\_activities DB default to 30, used for item #2 in the TAI

number\_of\_days\_after\_issued\_but\_not\_received DB default to 15, used for item #4 in TAI

number\_of\_days\_no\_payment\_issued DB default to 60, used for item #3 in TAI

number\_of\_days\_no\_payment\_after\_agreed: DB default to 15, used for item #6 in TAI

number\_of\_days\_AR\_outstanding DB default to 30, not used for now

number\_of\_days\_for\_delayed\_bill\_generation DB default to 90, used for delayed bill generation

number\_of\_days\_for\_litigated\_followup DB default to 90, used for item #5

number\_of\_days\_bill\_has\_not\_been\_generated DB default to 2, used for item #8 in TAI

number\_of\_days\_after\_offer\_but\_no\_agreement. Need to add for 1.0, need to DB default to 2, used for item #7 in TAI

close\_to\_claim\_filing\_deadline DB default to 10%/0.1, not used for now

invoice\_deliver\_preference DB default to Email., not used for now

reports\_delivery\_preference DB default to Email, not used for now

optioscol: not sure what this is

option\_name: not sure what this is

default\_place\_of\_servivce DB default to 24, place of service on the dropdown list. When not set, display “Select” for biller to select from the dropdown

# Workflows Scenarios

### Scan documents into PDF files on Windows

Use HP All-In-One device, or any other scanner able to scan to PDF, to scan document to PDF file on the Windows machine. For Service documents, we can scan a batch of cases into one PDF file as long as they all have the same number of pages per claim. The user can then use <http://www.pdfsam.org/> to split the single pdf file into one PDF for each claim. Make sure the split software is set not to compress the output files, which need to be pdf version 1.5 or later.

### Entering new service/claims for an existing patient

The biller enters the last name of the patient and at the end of the list of services/claims for the matched patient, there is an option to enter new service/claim for the same patient.

The biller should take a quick look at Patient and see whether any demographic info has changed since the last visit. If so, update the record

The biller should be taken directly to Services UI page to enter the new service data and save the claim. Biller also needs to scan and input the facilitydatasheet, ID and anesthesiatimesheet into the system.

The biller needs to double check to make sure there is no change in the existing patient’s demographic and insurance information before entering the new claim.

The biller saves the new claim at the Service UI page.

### Enter new patient/service/claim

The biller enters the last name of the new patient. If there is no one with that last name, the biller should be taken directly to the Patient UI page to enter a new patient. There is the option to enter a new patient at the end of matched list if there are matches with that last name.

### Receive a EOB and payment by mail

The biller uses any of the claims search criteria to find the claim that the EOB/payment is for in the system.

From the Claim Follow-up page

1. The biller scans and inputs the EOB and payment documents into the system. The system will not allow closing a claim without EOB.
2. Enter the EOB information to Claim Follow-up page under EOB.
3. Enter amount received
4. Change Claim Status.
5. Finish Claim

### Received additional payments from insurance

Enter the additional amount to the amount received and added an entry to log to indicate additional payment received.

### Received a EOB, but not payment

The biller uses any of the claims search criteria to find the claim that the EOB/payment is for in the system.

From the Claim Follow-up page, the biller scans and inputs the EOB document into the system.

Biller enters the payment to “Amount insurance payment issued” and the “Date insurance payment issued” as will all be automatically populated with today’s date.

The biller changes Claim Status to “Open – payment issued not received” and check the UI for “is the payment sent to patient” if that was the case. Also enter the EOB data on Claim Follow-up.

### Received a payment by direct deposit

Biller goes to Claim Management|Insurance and lookup the instruction on how to retrieve the EOB. Enter the payment in Claims Follow-up and into Document Management.

### Received a fax offer for a claim and payment agreement

Biller finds the claim and enters the “insurance initial offer”, the system will populate the “Date of Initial Offer” with today’s date.

After agreement was reached, the biller enters the “insurance agreed payment” and the system populates “Date of Agreement” with today’s date. Also input the agreement document.

This claim will be in TAI if agreement is not entered into the system in 2 days.

### Received a EOB with denial

Input the DOB document into the system and change the Claim Status based upon the reason for the denial

### Received a request from payer for W9

Go to the Claims Management|Provider and open the W9 file and print and fax. Make an entry into the log for the claim.

### Receive a request from insurance that a change is needed in the claim

The biller finds the claim, makes the change and rebill as prompted by the system. If no rebill is needed, answer no to the prompt.

### TAI processing

The biller starts with the claims high on the list of TAI list. Depending on the reason why this claim is on the TAI list, the biller will update the claim to get the claim removed from the list. If the processing of the claim requires a call to the insurance, the biller will get the insurance phone number and sort the TAI list by payer, and it should gather all claims that are on TAI for the same payer so that when contact is made to the payer’s claims department, all claims on the TAI list for the same payer can be discussed on the one call.

# Document Templates

We use a combination of PDF overlay and text string with variable substitution to compose the fax cover sheet and statements.

The cover sheet for payer type LI will be a little different from other payer types so I created two templates at <https://bitbucket.org/ebsllc/billingsystem/issue/1/fax-cover-sheet-template>

## Fax Cover Sheet

Fax Cover Sheet-LI.docx and Fax Cover Sheet-Non-LI.docx

## Type I Statement

StatementTypeI.dotx

Billing Company Name, address, phone number and fax number  
Name:patient last\_name, first\_name  
Street : patient.street\_address  
CityStateZip: patient cit state zip  
DOS: encounter start\_date\_1  
Amount: claim EOB\_allowed\_amount  
Insurance sent to you: claim amount\_insurance\_payment\_issued  
Your Responsibility: claim EOB\_allowed\_amount  
TitleName: Mr. or Ms. patient last name, depend on patient sex  
ProviderName: provider provider\_name  
RenderingProvider: renderingprovider first\_name last\_name  
Facility: facility facility\_name city state

## Type II Statement

StatementIITemplate.dotx

Variables are the same as type I

## Type III Statement

StatementIIITemplate.dotx

Variables are the same as type I

## Provider Report

SampleReport.docx

ProviderName: Provider.provider\_name

# DB Related Details and Changes

### DB field length

Street Address VARCHAR(60)  
City VARCHAR(60)  
State VARCHAR(20)  
Zip VARCHAR(9)  
Name VARCHAR(60) first and last name together  
Last Name VARCHAR(40)  
First Name VARCHAR(40)  
Company Name VARCHAR(60)  
Diag Code VARCHAR(8)  
CPT Code VARCHAR(10)  
NPI VARCHAR(11)  
Tax ID BARCHAR(9)  
Diagnosis Pointer VARCHAR(8) possible value “1”,“1,2”,“1,2,3”,“1,2,3,4”  
Log Text  
Notes Text   
Insurance ID VARCHAR(20)  
phone/fax/cell number VARCHAR(10)  
EDI VARCHAR(10)  
Claim Submission Preference VARCHAR(10) “Fax”,“Mail”,“EDI”  
File path to documents VARCHAR(200)  
Insurance Policy Group VARCHAR(20)  
Currency DECIMAL(8,2) ###,###.##  
Anesthesia rounding VARCHAR(20) “Always round up”, “Always round down”, “Closest whole number”, “No rounding”  
Provider Billing Rate ## Percentage  
Invoice/Report Delivery Preference VARCHAR(10) “Fax”,“Email”,“Mail”  
Signature 30 “Signature on file”  
Email address VARCHAR(60)  
Percentage Decimal(2,2)

### Accendo Account for DB connection

bsdb | bsconnect

### DB field values

Insurance.payer\_type

MM: Major Med  
ME: Medical  
PI: Personal Injury/Auto related  
WC: Workman’s Comp  
LI: Litigation   
SP: Self Pay

Insurance.claim\_submission\_preference

Mail  
 FAX  
 EDI  
 SelfPay

### Create some DB Views to help with

1. Claims that are open and billed. This might allow the TAI query to execute much faster.
2. All Claims to allow easier ad hoc query

### DB fields with default values

Encounter.accept\_assignment default to 1  
options.

### Pending DB fields changes

Need to record the insurance initial offer amount and date. This is also for this to be in TAI

Need to add to options number\_of\_days\_after\_offer\_but\_no\_agreement

Change options.provider\_invoice\_rate to decimal(2.2) Done

Change options.PIP\_rate to decimal(5.2). Done

Change claims.number\_of\_days\_after\_issued\_but\_not\_received default to 5 from 15

Change provider.billing\_phone\_number to not Null

Add options.number\_of\_days\_offered\_but\_not\_agreed

### DB structure changes need to be made on production

Add table zip2citystate

Replace table facility as we added 4 fields for Document Management

### Making DB schema changes without impacting data

Use mySQL TOAD <http://www.quest.com/toad-for-mysql/> to compare the original and updated schemas to confirm the changes are as intended and document the changes. Generate the sql file that makes the change and execute the change on the target machine and compare again. This time the schemas should be the same. I can then send out the sql file which can be executed from phpMyAdmin by others to sync the DB schema.

### DB Schema Changes

#### DB Schema Changes 2011 11

Included:

1. Standardize on field length and type  
2. Check on nullness  
3. Add all of the fields that we think we need in version 1.0 to this point

* 1. Fields related to new UI items Hao Wei is adding
  2. Other fields

Excluded:

1. Any fields deletions. We will have to do them at some point before version 1.0. I have noted them as “To be removed…”. The only exception is
2. Any field name changes. There is one I would like to change, will do that at a later date. I have noted them as “To be changed…”
3. DB changes related to making Facility and Referring Provider per Provider, which will be made at a later date

**New tables**

Added zip2citystate table to support zip to city and state lookup

Added placeofservice table to support Place of Service dropdown on Service UI

You have to get these tables and their data from bitbucket if you do not have them already

**Structure Changes:**

Biller

Change notes from varchar(60) to Text

Claim

Changed all benefit\_\* from varchar(45) to decimal(8,2) as they are CURRENCY.  
Changed benefit\_co-insurance from varchar(45) to decimal(2,2) as this is percentage  
  
To be Changed EOB\_deductable to EOB\_deductible”   
To be Removed file\_path\_to\_EOB\_image, file\_path\_to\_check\_image  
To be removed: is\_the\_issued\_payment\_ok. Date\_statementII, notes

Cptcode

Changed description from null to not null

Encounter

Change diagnosis\_pointer\_1 to not null   
Change accept\_assignment from TINYINT(4) to TINYINT(1)  
Added date\_same\_illness, not\_able\_to\_work\_from\_date, not\_able\_to\_work\_to\_date  
Added hospitalization\_from\_date, hospitalization\_to\_date, outside\_lab, charge  
Added data set 3 to 6

To be Removed file\_path\_service\_sheet and file\_path\_anesthesia\_time\_sheet\_image

Facility

Added service\_doc\_first\_page, service\_doc\_second\_page, service\_doc\_third\_page and service\_doc\_forth\_page to support Document Management to split the Services sheets

Followups

Added log entry data set from 5 to 8.

Changed Negotiated\_payment\_amounted from varchar(45) to decimal(8,2)

Added date\_initial\_offer and amount\_initial\_offer to support TAI # 7

To be Removed file\_path\_to\_negotiated\_agreement

To be removed: claim\_id and claim\_is\_inactive

Insured

Some changes to other\_insured\_\*

To be removed file\_path\_to\_ID\_card

Modifier

Changed modifier and description to not null

Options

Changed number\_of\_days\_no\_payment\_issues from not null to null

Patient

To be removed insurance\_card\_image

Provider

Changed billing\_\* to not null

Surgeryanesthesiacrosswalk

Changed description to not null

# Actions Triggered by DB Events

#### New claim created

Set the claim.claims\_status to Open\_not\_billed or open\_delayed\_bill\_generation. Fill in claim.date\_creation with today’s date.

#### Existing claim updated

Rebill logic as described in Bill Generation section

#### Claims.amount\_insurance\_payment\_received is updated

Update claims.amount\_insurance\_payment\_issued to the same amount if it is empty. Also populate Date insurance Payment Received with today’s date

#### Claims.amount\_insurance\_payment\_issueded is updated

Populate Date Insurance Payment Issued with today’s date. Update claim.claim\_status to Open- payment\_issued\_not\_received if claim.amount\_insurance\_payment\_received is empty

#### Bill is generated

Set claims\_status to either Open-billed or Open-rebilled based upon the logic in Bill Generation section. Set claim.date\_billed to today’s date

#### StatementI is generated

Set claim\_status to Open- payment\_issued\_not\_received\_check\_requested

#### Send to Patient is checked

Set claim.statement to I to get ready for a type I statement to be generated

#### Claim is set to closed-\*

Update the claim.date\_closed with today’s date

#### Claim is closed with payment as expected

The system find all claims that are still in Open\_delayed\_billing and ask the biller whether to make them Open-Not\_billed.

This is so that these claims can get billed ahead of their delayed billing time expires

# Productionization

|  |  |  |  |
| --- | --- | --- | --- |
| **Role** | **URL** | **Machine** | **Responsible Person** |
| Dev |  | Laptop | Hao Wei |
| Dev |  | Laptop | Xinwang Qiao |
| QA | 192.168.1.3/billingsystem | Ultra-win7 | Lin Chen |
| Production (version 0.9) | Bs.servebeer.com/billingsystem | Fileserver | Lin Chen |
| Production (version 1.0) | 192.168.1.20/billingsystem | Appserver | Lin Chen |

10/10/11: Started using 0.9 on fileserver

10/27/11: Copied code from QA to file server

# Main Page UI Layout

#### Claim Management (available to all billers)

Today’s Action Items  
 Claim Search  
 Bill Generation  
 Patient Statements  
 Reports

Insurance  
Facility  
Provider  
Rendering Provider  
Referring Provider  
Billing Company

#### Data Management (only available to billers with Bill Company admin right)

Patient  
Service  
Insured  
  
Facility  
Provider and Options  
Rendering Provider  
Referring Provider  
Billing Company

#### System Management (only available to billers with System admin right)

Billing Company  
Logon ID and rights assignment   
Insurance

Diagnosis Codes  
CPT Codes  
Anesthesia Crosswalk

## Claim Management

These functions are available to all billers.

The second group of data is read only, which is used for looking up data on these tables and also to access associated documents. We need some visual cue so the biller knows these data are read only

## Data Management

These functions should only be available for a Billing Company admin biller. These functions need to be able to prompt the user for the linked data. The data that the Billing Company admin can modify are obviously limited to data under the Billing Company

#### Patient, Service and Insured

Search and delete

Facility/Provider and Options/Rendering Provider/Referring Provider

Add, update and delete through a dropdown. These are all related to this Billing Company only

## System Management

These functions are available only to system wide administrators. These functions need to be able to prompt the user for the linked data.

#### Billing Company/Logon ID and rights assignment/Insurance

Add, update and delete through a dropdown

#### CPT/ Anesthesia Crosswalk/Diagnosis codes

Add, update and delete through a dropdown

# Main Claim UI Data Validation and Click Order

## Nullness enforcement

We will check null values as the biller leaves the UI page. We need to let the biller know that a value is left at null and it cannot be null. Please use phpMySQL to check whether null is allowed in the DB tables.

## Common data type restriction

Date: ##/##/####, where ## is <= 12, ## is <= 31, #### is > 1900 and <= current year

Time: ##:## ## is <= 23 and ## is <= 60.

## Patient

|  |  |  |
| --- | --- | --- |
| **Data** | **Restriction** | **Null** |
| Last Name |  | N |
| First Name |  | N |
| Date of Birth | DATE | N |
| Sex | Dropdown | N |
| SSN | ###-##-#### | Y |
| Street Address |  | N |
| Zip code | #####-#### | N |
| State |  | N |
| City |  | N |
| Phone Number | (###)###-##### | N |
| Second Number | (###)###-##### | Y |
| MRN |  | N |
| Patient Status | Dropdown | Y |
| Relationship to Insured | Dropdown | N |
| Is Patient Condition Related to | Dropdown, multiple selects | Y |

Last name, First name, DOB, Sex

Street address, Zip, City, State,

SSN, Phone number, Secondary phone number, MRN

Patient Status, Relationship to Insured, Is Patient Condition Related To

Notes

## Insurance

|  |  |  |
| --- | --- | --- |
| **Data** | **Restriction** | **Null** |
| Last Name |  | N |
| First Name |  | N |
| Date of Birth | DATE | N |
| Sex | Dropdown | N |
| SSN | ###-##-#### | Y |
| Street Address |  | N |
| Zip code | #####-#### | N |
| State |  | N |
| City |  | N |
| Insurance Type | Dropdown | Y |
| ID Number |  | N |
| Group Number |  | Y |
| Employer Name |  | Y |
| Is there another health benefit plan? | Checked or not checked | N |
| Insurance Company Info | Entire content not changeable | N |

Last name, first name, DOB, sex

Street address, zip, city, state,

SSN, Phone number, secondary phone number

Insurance type, Insurance ID, Group Policy or FECA Number, Employer Name or School Name

Is there another Health Benefit Plan? If this is checked, need to display the items below

(Other Insured Last Name, First Name, DOB, Sex

Other insured policy or group number, Employer Name or School Name, Insurance Plan Name or Program Name)

Notes, for insured

Company Name, Phone Number, Fax Number,

Street address, City, State, Zip

Phone extension for claims, phone extension for benefits

Payer Type, EDI number, Claim submission method, anesthesia billing rate, anesthesia crosswalk overwrite

Benefit lookup, claim lookup, EFT lookup, claim filing deadline,

Reconsideration, appeal, navinet support number, PID

Notes, for Insurance

## Service

|  |  |  |
| --- | --- | --- |
| **Data** | **Restriction** | **Null** |
| Date of Current Illness or Injury | DATE | Y |
| Date Patent has had same or similar illness | DATE | Y |
| From date patient unable to work in current occupation | DATE | Y |
| To date patient unable to work in current occupation | DATE | Y |
| Hospitalization related to current services, from date | DATE | Y |
| Hospitalization related to current services, from date | DATE | Y |
| Outside lab | Check or not checked | Y |
| Charge | CURRENY | Y |
| Medicaid resubmission code |  | Y |
| Original ref number |  | Y |
| Prior authorization number |  | Y |
| Facility | Dropdown | N |
| Provider | Dropdown | N |
| Rendering Provider | Dropdown | N |
| Referring Provider | Dropdown | Y |
| Diagnosis code 1 | Dropdown | N |
| Diagnosis code 2 | Dropdown | Y |
| Diagnosis code 3 | Dropdown | Y |
| Diagnosis code 4 | Dropdown | Y |
| Start Date 1 | DATE | N |
| End Date 1 | DATE | N |
| Anesthesia Start Time 1 | TIME | N |
| Anesthesia End Time 1 | TIME | N |
| CPT code 1 | Dropdown, multiple selects | N |
| Anesthesia Minutes 1 | Derived, not changeable | N |
| Place of Service 1 | Dropdown | N |
| Crosswalk 1 | Derived, changeable | Y |
| Modifier 1 | Dropdown, multiple selects | Y |
| Anesthesia Unit 1 | Derived, changeable | Y |
| Diagnosis code pointer 1 | 1/1,2/1,2,3/1,2,3,4 | N |
| Charge 1 | CURRENCY, derived not changeable | N |
| EMG | Dropdown of select/yes/no | Y |
| EPSDT/Family Plan | 1 letter | Y |
| Accept Assignment | Dropdown | N |

Date of current illness of injury, Date patient has had same or similar illness, Dates patient unable to work in current occupation – from date, to date

Hospitalization related to current services – from date, to date, Outside lab? Charge

Medicaid resubmission code, Original ref number, Prior authorization number, Accept Assignment

Facility, Provider, Rendering provider, Referring provider

Diagnosis codes 1, 2, 3, 4

Star t Date 1, End Date 1, Start Time 1, End Time 1

CPT Code 1, Modifier 1, Crosswalk code 1, Charge 1

Anesthesia mins 1, Anesthesia units 1, Place of service 1, Diagnosis Code Pointer 1,

EMG 1, EPSDT/Family Plan 1

(Date sets 2-6 are the same as date set 1 above, but they can all be NULL. They should only display if they contain data)

Notes

## Claim Follow-up

|  |  |  |
| --- | --- | --- |
| **Data** | **Restriction** | **Null** |
| Total Charge | CURRENCY, derived not changeable | N |
| Payment Expected | CURRENCY, derived, changeable | Y |
| Claim Status | Dropdown | N |
| Date bill created | DATE, derived, changeable | N |
| Date Billed | DATE, derived, changeable | Y |
| Date closed | DATE, derived, changeable | Y |
| Insurance payment Issued amount | CURRENCY | Y |
| Insurance payment Issued date | DATE, derived, changeable | Y |
| Insurance payment received amount | CURRENCY | Y |
| insurance payment received date | DATE, derived, changeable | Y |
| Sent to patient? | Check or not checked | Y |
| Date and Time | DATE TIME | Y |
| Insurance proposed amount | CURRENCY | Y |
| Insurance proposal date | DATE, derived, changeable | Y |
| Insurance agreed payment | CURRENCY | Y |
| Insurance agreement date | DATE, derived, changeable | Y |
| Date of next follow-up | DATE | Y |
| EOB entries | CURRENCY | Y |
| Benefit entries | CURRENCY | Y |
| Benefit\_co-insurance | 1-99 to represent percentage | Y |
|  |  |  |
|  |  |  |
|  |  |  |

Total charge, Amount paid, Balance due, Payment expected

Claim Status, Date of next follow-up

Date bill created, Date billed, Date last billed, Date closed

Insurance payment Issued amount, Insurance payment issued date, send to patient?

Insurance payment received amount, insurance payment received, Date last checked with attorney

EOB: Allowed amount, not allowed amount, Co-insurance, Deductable

Benefit: OOP, Remaining, Deductible, Remaining, Co-Insurance, date

Date and Time 1, Who/Number/Ref Number 1, Log 1

(We need to have 8 total, but only display the first one or the ones with data)

Insurance proposed amount, insurance proposal date, insurance agreed amount, insurance agreement date

Notes

# Source Code Documentation

We will use Doxygen to document our code

Download Doxygen from <http://www.stack.nl/~dimitri/doxygen/download.html#latestsrc> and DOT from <http://www.graphviz.org/Download_windows.php>. Use the configure file in docs folder under Doxygen and make changes to Source code directory and destination directory to fit your need. Destination is where the generated HTML file should be placed.

Once you have the configure file correctly configure, just run doxygen and use the path to the config file as the only parameter.

I have placed the HTML files at bs.servebeer.com/Doxygen/html. You can take a look and I will update it from time to time.

We will use Javadoc style document to comment the code. Please see <http://www.stack.nl/~dimitri/doxygen/docblocks.html> and see the example towards the middle of the page

# Automated Regression Testing

We use Selenium IDE at seleniumhq.org. Please download it from <http://release.seleniumhq.org/selenium-ide/1.7.2/selenium-ide-1.7.2.xpi> and install it as a Firefox extension.

You need to use Demo DB and this test will login as hmb-admin.

Open Selenium IDE from Firefox |Tools | Selenium IDE.

File | Open and select AddTestUsers.html. You might need to make some changes:

1. Set Base URL to your web server, most likely <http://localhost/>. Do not add billingsystem/public to the end, only the web server name.
2. Slide the playback speed slider from Fast to Slow.
3. Make changes in the Parameters section at the top of the Table if you need:
   1. ServiceDoc the path to the PDF file that is provided to Service UI Page
   2. FirstUser, SecondUser and ThirdUsers are the test user names this test will add. You need to make sure you either do not have any of these users in your DB, or you modify the parameters to user a different set of user names

Click the icon for “Play Current Test Case” and this test will execute. You can let it run by itself and come back when it finishes to check the result. If there is any error, you can see the specific error from Log at the bottom of your Selenium ID window.

Please note that I was not able to interact with the Document Management popup after Finish Claim at this point, even though it suppose to work. What I have the script do now is to up the Document Input URL in the main window. You can either close the popup as they come up during the play back or close them without impacting the test cases.

If we grow out the capability of Selenium because our tests are too complex and need more logic in the test cases, we need to look into Selenium WebDriver or maybe saucelabs.com.

# Source Code Management

## NetBeans Mercurial Setup and Configuration

Install HG and Mercurial from <http://mercurial.selenic.com/downloads/> using TortoiseHg with Mercurial x64 or x86 depending on your OS.

Go to NetBeans|Tools|Options|Miscellaneous|Versioning|Mercurial and change Mercurial User Name to “Xinwang Qiao [qiaoxinwang@gmail.com](mailto:qiaoxinwang@gmail.com)”. I had to change the Mercurial install directory from c:\Program Files\Mercurial to C:\Mercurial as NetBeans complains about the space between Program and Files so my Mercurial Executable Path says C:\Mercurial\bin\hg.cmd

Connect to Bitbucket.org and login using ID and password below:

[ebsllc@hotmail.com](mailto:ebsllc@hotmail.com)  
linchen19 | qwerty  
omarhao | qwerty  
qiaoxinwang|qwerty

There are three entities in Mercurial

1. Working Directory. This is the copy you are working with in NetBeans.
2. Local Repository. This is the Mercurial repository that is stored on your local computer. We have it in the same folder as your Working Directory under a folder by the name of .hg. This repo holds all of your changes that you have committed and you can rollback to.
3. Remote Repository. Shared repo on bitbucket.org

Working Directory can commit changes to Local Repo. Push and pull/fetch are used between Local Repo and Remote Repo.

Start NetBeans, close all project so NetBeans is empty. Go to Team|Mercurial|Clone other

Put in http://bitbucket.org/ebsllc/billingsystem for Repository URL and use the ID and password above. Next and confirm http://bitbucket.org/ebsllc/billingsystem as default pull and push path. Next and select the Parent Directory, or use the default and Clone Name will be the folder under the parent, so keep billingsystem and click Finish. This is when the entire source code will come down. It only take a a few mins for me, but might be longer for others.

Open it up and make some changes. Click Team|Commit and type a line that summarizes what this change is about and Commit. This will Commit the changes you made into the local repository, which is in .hg folder under test folder that you selected when you Clone.

At this point, you can sync your changes back to bitbucket using Team|Share|Push to Default. The other person can do the clone and will get the updated copy.

Once you both have your local copy, you will need to do Team|Share|Pull and Team|Merge|Merge Changes to merge changes into your local repo.

If you can not push to bitbucket, make sure your NetBeans knows where your Mercurial is installed. The newest NetBeans 7.01 seems to have an issue, so use the one in D:\Data\Docs\Accendo\Version Control and do not update.

Need to make sure we copy, not cut, c:\program files\mercurial to c:\mercurial.

Always select the Root Source file before you execute any command for the project.

## Commonly used Mercurial commands in NetBeans

Status show files with uncommitted changes in Working Directory. Right click the file allows you to see the changes within the file.

Diff Show differences between the Local and Remote Repo

Update Sync changes from Local Repo to Working Directory

Commit Sync changes from Working Directory to Local Repo

Clone Copy the entire Remote Repo to Local Repo and Working Directory. This should only be used for the initial download.

Fetch Pull, update and merge to sync Remote Repo changes to Local Repo and Working Directory

Share | Push Default Sync changes from Local Repo to Remote Repo

Share | Pull Default Sync changes from Remote Repo to Local Repo

#### Revert Modification

Revert Working Directory to a previous revision. You need to first select file that you want to revert modification.

#### Recover | Rollback

## Workflow

1. Everyone clone from the Remote Repo
2. Person 1 makes changes in Working Directory and test. After done changes, commit to Local Repo.
3. Person 2 makes changes in Working Directory and test. After done changes, commit to Local Repo.
4. Person 1 does a fetch from Remote Repo in case person 2 might have pushed some changes to Remote Repo. A push follows that pushes all of the changes to Remote Repo.
5. Person 2 does a fetch from Remote Repo and merges the person 1 changes with the local changes on Local Repo and Working Directory. A push follows that pushes the combined changes to Remote Repo
6. Person 1 does a fetch from Remote Repo and get the combined changes to Local Repo and Working Directory
7. At the point, the two persons have the code in sync again on the Remote Repo and ready for more changes.
8. To back out changes from Working Directory, use Revert Modification. This leaves the Local Repo unchanged.
9. To back out a commit from Local Repo
10. To back out a push from Remote Repo

Keep in mind that before you push, you should **always do a fetch first.** If you do a push without a fetch/pull first and others had done a push after your last fetch/pull, you will get a message that warns you will create a branch on the Remote Repo.

## Mercurial and Bitbucket References

<http://hgbook.red-bean.com/read/index.html>

<http://confluence.atlassian.com/display/BITBUCKET/Bitbucket+101>

<http://jemander.se/MercurialByExample.pdf>

<http://netbeans.org/kb/docs/ide/mercurial.html>

# Test Plan

#### Claim Search

Search claims uing each of the search criteria and combinations of them. Use

#### Enter new claim

Enter a new claim by searching a non-exist last name as well as by

#### Updating an existing claim

#### Claim follow-up

#### Bill Generation

Mail

Fax

EDI

#### TAI

Change options table

# Open issues need solution

#### SCM

A source code management system that works well with limited connectivity.

Mercurial and Bitbucket.org?

### Scanning Integration

Morena  
JTwain   
Dynamic Twain  
No cost Java image acquisition framework <http://www.java.net/pub/au/174>

To get started, we can use HP’s scanner utility to scan all of the pages for a services in a day into one PDF file, and then let the PHP code to take the file and separate into individual files by page and then store them into the right folder and the file names for the system to pickup. We are assuming each service is made up of three pages. The PHP code just needs the MRN numbers for the patient and it will place the files in the right folder for the patient. The system needs to know the MRN of these patients.

### Document acquisition UI design and one-off documents

Should we place the document acquisition UI/button next to the access button, which are located at the UI pages that the documents are relevant for or we should have the document acquisition UI in one separation area of the system together for all document?

How do we deal with documents that are not standard?

#### Fax Integration

Via email, using SMTP transport from hotmail.com or gmail.com. We will consider running our own SMTP transport on synology box in 2.0. <http://www.markinthedark.nl/news/ubuntu-linux-unix/83-configure-mailstation-2-on-synology-ds211j.html>

Spoke to Naj at myfax on 10/19/11. Leaving the email subject and body blank will make sure the only content gets sent is in the PDF file attached to the email. The top line on the fax that has the business name and fax number is configured with the account.

We currently have pfax 0/1000 per month for $66/Y. We upgraded to 100/1000 at $88/Y 10/28

#### Ad Hoc query

How can we do to facilitate a practical means to support ad hoc queries? We need

1. A view, or several views, that makes it to easy to construct a selection statement
2. A tool that allows read only access to DB and have some basic results formatting

#### Per insurance exception for TAI criteria

How to allow some insurance to have different criteria for TAI selection so that for example UHC has 15 days of no interaction to be on the TAI list, vs. the normal is 30?

Horizon’s TAI item #2 should be 10 days vs the others 30. We add a field in insurance, insurance.tai\_item2. The TAI list generation will need to check this and see whether to use the default value from options table, or to use the one from the insurance table.

#### Zip code to City and State lookup

<http://www.dreamincode.net/forums/topic/10862-zip-code-to-city-state/>  
<http://zipinfo.com/products/z5lte/z5lte.htm>  
<https://www.usps.com/webtools/htm/Development-Guide-v3-1.htm>  
Your Username is 756EBS003020  
Your Password is 566LA51YH625  
We will use the local table solution, sourcing the data from zipinfo.com. The table is zip2citystate

#### Temporarily remove an item from TAI for a short period of time

Have a button on the list to “hide” an item from TAI for 1-7 days.

#### Brower related issues

1. How to dictate to always open a PDF file externally
2. What kind of problem that it does not open if downloaded, but works fine if we write to a file
3. HTML5 and Drag & Drop.

#### Add Claim status to search results besides search condition?

#### Branch and rollback in Mercurial

Need to be able to branch so I can keep a 1.0 branch and a 9.\* branch

#### Receive additional payments for a claim that has been paid already

#### How to support manually entering a crosswalk code and let the system to look up the units and calculate the charge?

If we use a entry like 9999 01991 01991/3 3 in the surgeryanesthesiacrosswalk table and let the system to ignore 9999 and use 01991’s 3 base unit to calculate the charge, that should cover this

#### How to alert the biller this code is not being paid by an payer\

We can have a per payer fields that has the CPT pair so that if that’s what’s to be put into the claim, alert the bill this insurance does not pay for this combination.

#### How to allow the biller to easily confirm the patient comes back for new service still has the same insurance?

Each UI page displays the patient name, MRN, DOS and insurance name

#### If a biller changes an insurance company from a MM to a PI, even though the DB is changed, the claim status stays at delayed bill generation that was set for MM

#### Patient billing for deductable or self pay

Add options. patient\_billing, which can take the value of all, deductable, insurancepay or none

If all or deductable is set, when EOB deductable is entered statement I is set to generate the statement

If all or insurancepet is set, statement I is generated when sent to patient is checked

Add options.selfpay\_billing, which can the value of referingprovider or patient

If all or deductable is set, when EOB deductable is entered statement I is set to generate the statement

If all or selfpay is set, Bill Generation needs to generate statement I to the patient

If all or insurancepat is set, statement ! is generated when sent to patient is checked

We need different template for each of the above.

#### Needs to know whether a UI data item has been changed

We have three things on this at this point:

1. We need to know if ANY thing was changed, does not need to know what it was, so that we can prompt the biller when he is browsing away from the UI pages after making any changes.

2. We have to know which UI page/table was changes so that we know what we offer the user to rebill.

3. If we have an expensive way to know which data item is changed, we can probably not have to write everything out to DB when Finish Claim, only write the data that are changed.

Of the three, #2 is required, which also covers #1. #3 we are not sure whether it is the right thing to do, depending on the cost.

#### Secondary Insurance Billing

Cases that we will bill secondary insurance  
 1. The primary insurance has a deductable or c-insurance, we auto triggers a bill to secondary insuance that include the EOB from primary insurance  
 2. When a PIP claims run out of benefit fund, we will the secondary insurance with the EOB from PIP.

#### Record lock

Need to lock record to prevent two users from writing to the same record.

#### Work flow

Who is working on which problem

# References

## Openemr

To get it to work on synology, add the following to interface\globals.php

$webserver\_root = "/volume1/web/openemr-4.1.0";  
$web\_root = "/openemr-4.1.0";

## Zipinfo.com

Download Instructions  
Web Page: <http://data.zip-info.com/dataoct/z5lte/z5lte.htm>  
User Name: PSTFAC  
Password: DTWHCT

Remove the headers and footer from the txt file. Use Gsplit (<http://www.gdgsoft.com/gsplit/>) to split into 10,000 lines each using pattern of 0x0D0x0A. You should end up with 5 GSD files.

To upload, empty the zip2citystate table, and remove ID field. Import as CSV file with “,” as field terminator. Do this for each of the GSD files. Each import will take 5 mins.

Add id field back as unsigned and auto increment.

I currently have the 10/2011 version with about 41,000 records.

## mySQL Log

add log="c:\genquery.log" to mysql\bin\my.ini under [mysqld]

#### LCD list needs to be upgraded

Reboot the printer

#### HTTPS

<http://answers.yahoo.com/question/index?qid=20080514113444AAMMt5t>

<http://roshanbh.com.np/2008/05/redirect-browser-https-ssl-php.html>

<http://www.php.net/manual/ro/function.header.php#83448>

<http://stackoverflow.com/questions/85816/how-can-i-force-users-to-access-my-page-over-https-instead-of-http>

<http://www.besthostratings.com/articles/force-ssl-htaccess.html>

<http://answers.yahoo.com/question/index?qid=20080514113444AAMMt5t>

How to reset MySQL password

<http://forums.zend.com/viewtopic.php?f=8&t=62&start=20>

To install Zend Server CE (PHP5.3), first stop XAMPP. Install from <http://www.zend.com/en/products/server-ce/downloads>. Select Custom install and add Zend Loader, phpMyAdmin and mySQL. Also make sure install apache is selected in the next screen. Also make sure port 80 is not being used by anything else to allow the Zend Server Apache to use it.

After install, edit http.conf so that AllowOverride All is present for the Web root folder.

You can go back and forth between Zend Server and XAMPP by starting and stopping Apach2.2-Zend service and MyXQL\_ZendServer51 and XAMPP Control Panel and kill mysql32.exe

Use Zend Server CE’s admin tool|Server Setup|Components|Zend Guard Loader|Directives to setup the path to the FPDI\_PDF-Parser’s license file, leave everything else the way they are. Restart PHP using the button at the lower right hand of the UI.

Change file permissions for the ini files so they are every one writable.

<http://veerasundar.com/blog/2009/01/how-to-change-the-root-password-for-mysql-in-xampp/>