1. Responsibility of Each Group Member

Frank Smith - Project Leader
Duties:

1) Proofread all work before it is submitted.
2) Write all weekly meeting minutes for group progress.
3) Remind all members of various due dates.
4) Write various sections of the projects documentation, and coding.

Member 1 - Project Coordinator
Duties:

1) Build all diagrams, tables and use cases.
2) Write project documentation, and various coding.
3) Build PowerPoint Presentation.

Member 2 - Project Developer
Duties:

1) Build various functions for the program.
2) Provide feedback, and written work for the project.
3) Build PowerPoint Presentation.

2. Meeting Minute Notes

Week of April 23rd, 2016

Coding issues are becoming present. While we are still aiming for a Monday completion date, testing will now begin on Sunday, May 1st, instead of April 30th. Corrections will be made on May 1st and May 2nd, and the completion date is scheduled for Tuesday, May 3rd. Writing the research paper will begin next Friday or Saturday, after the final presentation has been completed. The phase III paper is beginning to be completed, as various sections are trickling in from the designated members. Currently, I believe the project will be completed by the Wednesday, 10 PM due date.

Week of April 16th, 2016

I'm sorry I forgot to post this, I have so many projects going on that this slipped my mind. Last week, Member 1 updated all of the diagrams for phase III, and Member 2 was given the Graphical User Interface by me to complete the coding. All sections are on track for completion, but the coding section will be difficult with the two week constraint.

Week of April 9th, 2016

The group sections were being finalized, and the group met on Monday April 10th, to compile the document into its final form. The paper was posted on time, and complete with all sections needed for its submission.

Week of April 3rd, 2016

The group is currently running on time, and all tasks are being finalized. Based on the current schedule, the final deliverable for phase 2 will be distributed to all members on Monday evening, and any adjustments will be included on the posted document for Tuesday evening.

Week of March 27th, 2016

Currently everything is running on time - but due to the spring break, and the accompanying mid-term, phase II will be completed this weekend. We have designated various sections of the project, where Frank will be in charge of compiling the document with all of the various deliverables, as well as building the preliminary GUI for the program. Member 1 will build many of the various diagrams and models, and Member 2 is setting up the SQL server, and building the PowerPoint for the Presentation on April 7th. I believe we will run on time with the deliverable, and by having the due date pushed back, I have very little doubt that we will have any large problems.

Week of March 20th, 2016

Due to the spring break, and impending exam, no project work has occurred. On Saturday, March 26th, project sections will be sent out to all members, and the various project phase II sections will commence. So far, this team is working very well together, and I do not see any reasons for the project to fall behind on the timeline.

Week of March 6th, 2016

Group meetings were held on the 6th, 7th, and 8th to complete the Phase 1 deliverable, and prepare for the presentation. In addition, the group met early on the 10th to run through any questions before they were asked during the presentation.

We are on spring break for the following week, so there are currently no plans on meeting. We will need to meet virtually; however, since the research paper proposal is due on the 20th.

Week of February 21st, 2016

The team met with Dr. Song on Monday 2/29/2016, and provided our current materials for review. The team discussed the changes that needed to be made, and met

on 3/3/2016 to finalize the user stories, and build the DFD. We are currently on track, and need to build the PowerPoint once the document is finalized this weekend.

Week of February 21st, 2016

Our group met on Saturday the 20th, and completed the bulk of the phase 1 deliverable. Most of the written work has been completed, and it is currently being expanded upon. We have been in contact with each other throughout the week, increasing the completeness of the deliverable overall.

Our next meeting is at 4PM on 2/25/2016 to decide what additional tasks need to be worked on over the coming week.

Week of February 14th, 2016:

On **2/11/2016** the project leader sent the "Term Project Phase I" file to all members via Google Docs, providing everyone access to what is due, before class, on **3/8/2016.**

A group discussion, on the topic of the "Research Paper", was started February 15th via e-mail by the project leader. The topic that has been chosen is "Cloud Computing for Mobile Users". Each member will find an article to be evaluated by the team with a deadline of Saturday **2/20/2016** at 11:59 PM.

Following class on **2/18/2016**, team Binary will meet to discuss their strengths and weaknesses in relation to the given "phase I" requirements. Preliminary work will commence, and a group peer review will begin on **2/25/2016**. This will identify any issues that need to be improved, and provide help from other members when and if needed.

Table of Contents

1. Detailed Design 6
   1.1 Module detailed design 6
      1.1.1 Module 1 Detail – Login 6
      1.1.2 Module 2 Detail - Create Patient Account 8
      1.1.3 Module 3 Detail - Acquire Patient Information 10
      1.1.4 Module 4 Detail - File Claims 12
      1.1.5 Module 5 Detail - Manage Patient Account 14
      1.1.6 Module 6 Detail - Produce Invoice 16
      1.1.7 Module 7 Detail - Make Payment 18
      1.1.8 Module 8 Detail - Produce Patient Visit Summary 20
   1.2 Data Flow Diagram 22
2. User Manual 23
   2.1 Patient 23
      2.1.1 Logging into the System 23
      2.1.2 Patient Demographics 24
      2.1.3 Insurance 25
      2.1.4 Billing 26
      2.1.5 Primary Physician 27
      2.1.6 Patient Balance 28
      2.1.7 Patient History 29
   2.2 Doctor 30
      2.2.1 Logging into the System 30
      2.2.2 Main Menu 31
      2.2.3 Patient Form 32
      2.2.4 HCFA 38
      2.2.5 Create Patient Account 44
      2.2.6 Insurance Company 45
      2.2.7 Provider Form 46
      2.2.8 ICD-10 47
      2.2.9 Diagnosis 48
      2.2.10 Compact Database 49
   2.3 Administrator 50
      2.3.1 Logging into the System 50
      2.3.2 Main Menu 51
      2.3.3 Patient Form 52
      2.3.4 HCFA 58
      2.3.5 Create Patient Account 64
      2.3.6 Insurance Company 65
      2.3.7 Provider Form 66
      2.3.8 ICD-10 67
      2.3.9 Diagnosis 68
      2.3.10 Compact Database 69
3. Deployment Diagram 70
4. Implementation of User Story 70

4.1 All Users 70
4.1.1 Log onto the System 70
4.1.2 Search for Patient 71
4.1.3 Produce a Claim 71
4.1.4 Update Account Balance 71
4.1.5 View Account 71
4.1.6 Make a Payment 71

5. Test Case 71
5.1 Test Case #1 72
5.2 Test Case #2 73
5.3 Test Case #3 74
5.4 Test Case #4 75
5.5 Test Case #5 76
5.6 Test Case #6 77
5.7 Test Case #7 78
5.8 Test Case #8 79
5.9 Test Case #9 80
5.10 Test Case #10 81
5.11 Test Case #11 82

6. Traceability Analysis 83

1. Detailed Design

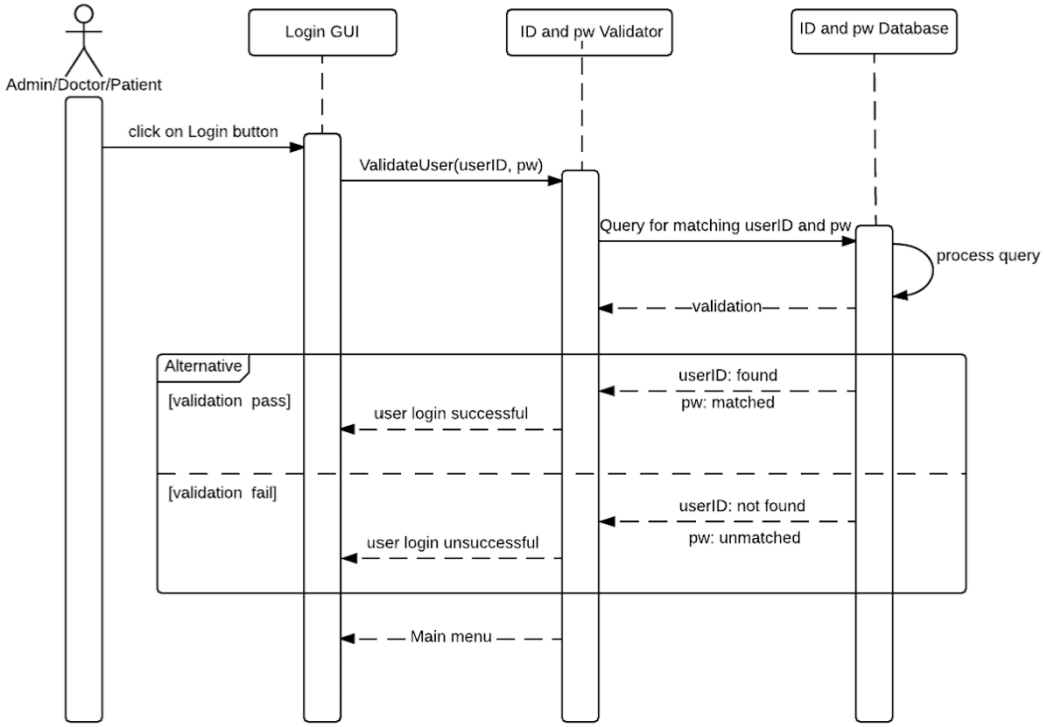
All included information has been enhanced, with the exception of the data flow diagram. This was included for reference when viewing the traceability table.

1.1 Module detailed design

1.1.1 Module 1 Detail – Login

Use Case Name: Login	ID: UC01
Primary Actor: Admin, Patient, Doctor	
Description: This use case describes how a user logs into the MedEasy system	
Trigger: The user clicks on “login”	
Precondition: The actor must be authorized	
Post condition: Login is successfully	
Basic Flow: 1. The system requests that the actor enter his/her ID and password. 2. The actor enters his/her ID and password. 3. The system validates the entered ID and password, and logs the actor into the system. 4. The use case ends.	
Alternative Flows: *1. Invalid Name / Password: If in the Basic Flow the actor enters an invalid name and/or password, the system displays an error message. The actor can choose to either return to the beginning of the Basic Flow, or cancel the login, at which point the use case ends.*	

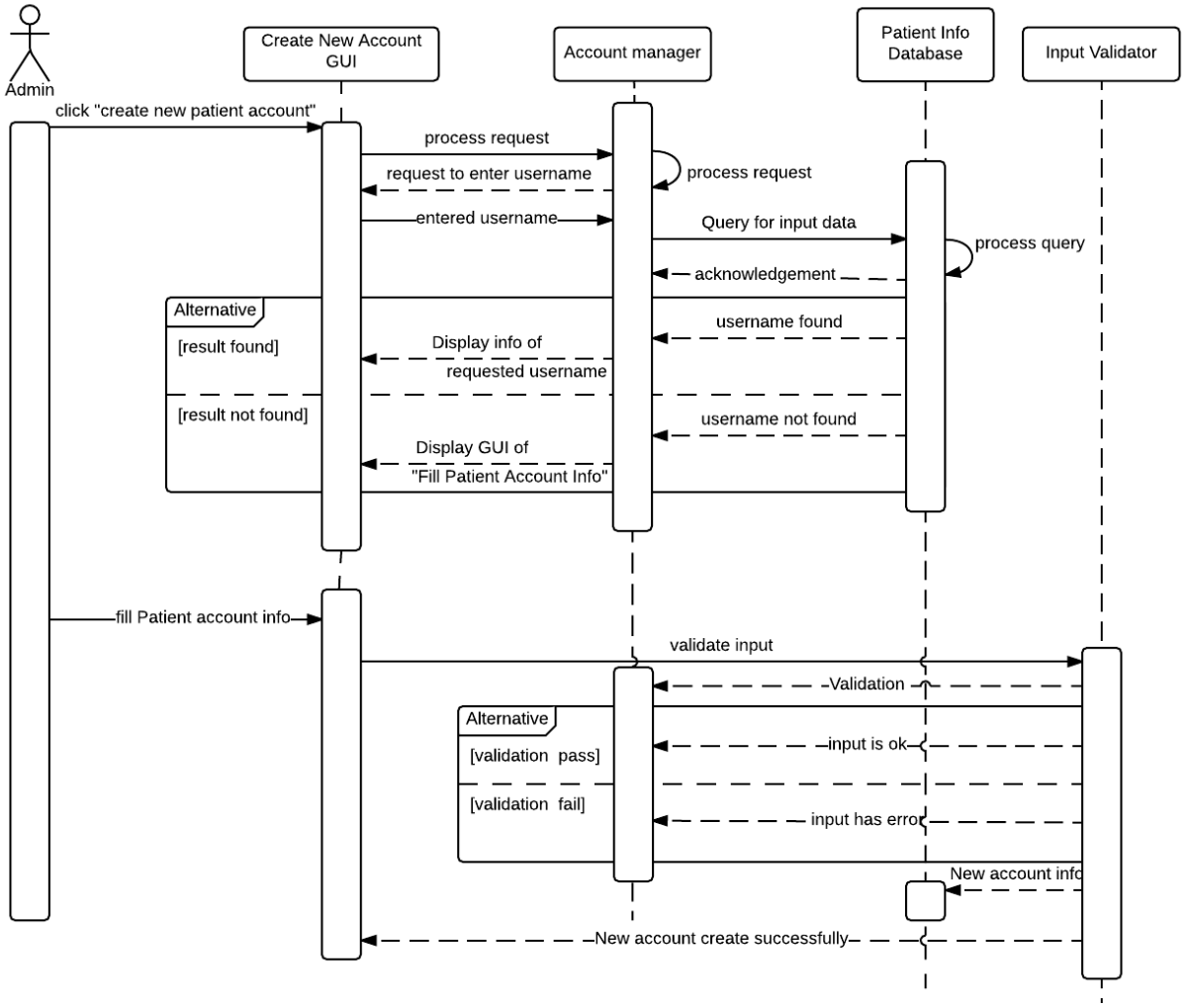
Figure 1


Admin/Doctor/Patient
Login GUI
ID and pw Validator
ID and pw Database
click on Login button
ValidateUser(userID, pw)
Query for matching userID and pw
process query
validation
Alternative
[validation pass]
userID: found
pw: matched
user login successful
[validation fail]
userID: not found
pw: unmatched
user login unsuccessful
Main menu

1.1.2 Module 2 Detail - Create Patient Account

Use Case Name: Create patient account	ID: UC02
Primary Actor: Admin	
Description: This use case describes how an admin creates a new account for a patient	
Trigger: The user clicks “Create New Account”	
Precondition: The actor must be authorized	
Post-conditions: The new account has been created successfully	
Basic Flow: 1. The actor enters patient ID. 2. The system’s database query if the username is existed. 3. The new account creation page will be displayed if the username is not exiting. 4. The system validates the entered Patient Account information. 5. The Patients Account information is stored in the Patient Info database. 6. The system displays a “New account created successfully” message. 7. The use case ends.	
Alternative Flows: 1. List User’s Information: If the searched username has been found from the database, the information of the user will be displayed. 2. User Enters Invalid Patient Account Information: During Create Account, if the system determines that the user entered invalid New Patient Account information, the following occurs: 1) The system describes which entered data was invalid, and presents the User with suggestions for entering valid data. 2) The system prompts the User to re-enter the invalid information. 3) The User re-enters the information and the system re-validates it. 4) If valid information is entered, the Patient Account Information is stored. 5) If invalid information is entered the invalid alternative flow is executed again. This continues until the User enters valid information, or chooses Cancel 6) Invalid Patient Account information includes: o Missing information items o Username already exists in the system o Patient Account information entered does not comply to its definition in the glossary o Improperly formed e-mail address	

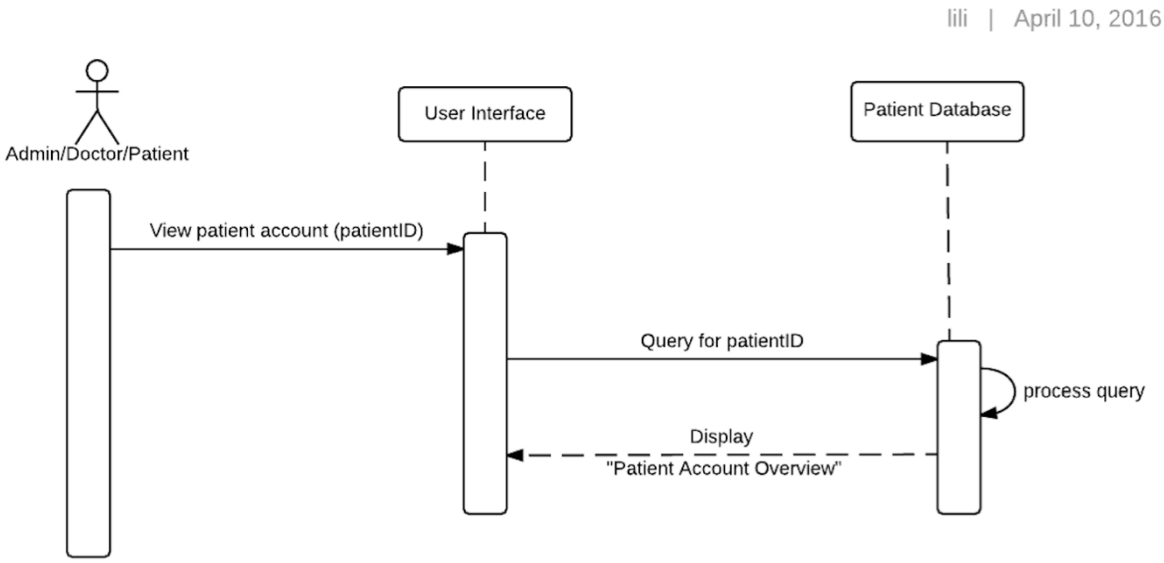
Figure 2


Admin
Create New Account GUI
Account manager
Patient Info Database
Input Validator
click "create new patient account"
process request
request to enter username
process request
entered username
Query for input data
process query
acknowledgement
Alternative
[result found]
username found
Display info of
requested username
[result not found]
username not found
Display GUI of
"Fill Patient Account Info"
fill Patient account info
validate input
Validation
Alternative
[validation pass]
input is ok
[validation fail]
input has error
New account info
New account create successfully

1.1.3 Module 3 Detail - Acquire Patient Information

Use Case Name: Acquire Patient Information	ID: UC03
Primary Actor: Admin, Doctor, Patient	
Description: This use case describes how an admin acquires a patients information from the system	
Trigger: The user enters the patient's ID	
Precondition: The user must be authorized, and the searching patient has to be an existing user	
Post condition: Acquired Patient Info successfully	
Basic Flow: 1. The actor enters the patient's ID in the search engine. 2. The system displays the search results. 3. The use case ends.	
Alternative Flows: 1. Search not found: If the actor entered an invalid patient ID, or non-existent patient ID	

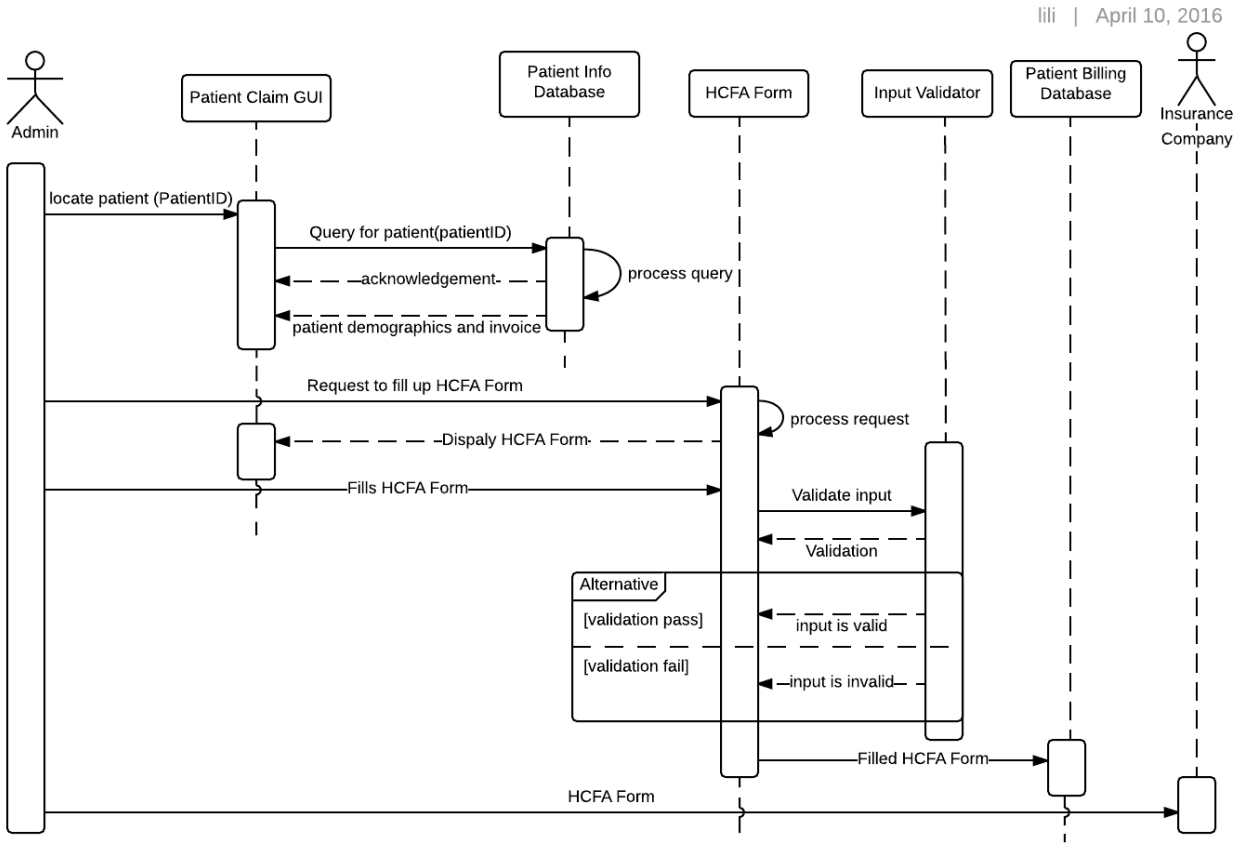
Figure 3


lili | April 10, 2016
Admin/Doctor/Patient
User Interface
Patient Database
View patient account (patientID)
Query for patientID
process query
Display
"Patient Account Overview"

1.1.4 Module 4 Detail - File Claims

Use Case Name: File claims	ID: UC04
Primary Actor: Admin Passive Actor: Insurance Company	
Description: This use case describes how an admin files claims for patients	
Trigger: The user clicks “File Claim”	
Precondition: The user must be authorized, and the patient’s invoice has been produced for his/her visit	
Post condition: Claims have been filed, and stored in the Patient Billing data store	
Basic Flow: 1. The actor search’s for the patient using the patient’s ID. 2. The system returns the search results, including the patients demographic and insurance information. 4. The actor fills the HCFA claims form out for the patient. 5. The claim form will be stored in the Patient Billing data store. 6. The actor sends the claim form to the insurance company. 7. The use case ends.	
Alternative Flows: *1. The actor enters invalid information in the claim form: If the system determines that the user entered invalid information, the following occurs:* 1) The system displays a message when a field is required. 2) The system prompts the User to re-enter the invalid information. 3) The user re-enters the information, and the system re-validates it. 4) If valid information is entered, the Claim Form(HCFA) is stored. 5) If invalid information is entered, the Invalid alternative flow is executed again. This continues until the User enters valid information, or chooses Cancel. 6) Invalid Patient Account information includes: - Missing information items - The claim form information that is entered does not comply to its definition in the glossary - Improperly formed information, such as an email address.	

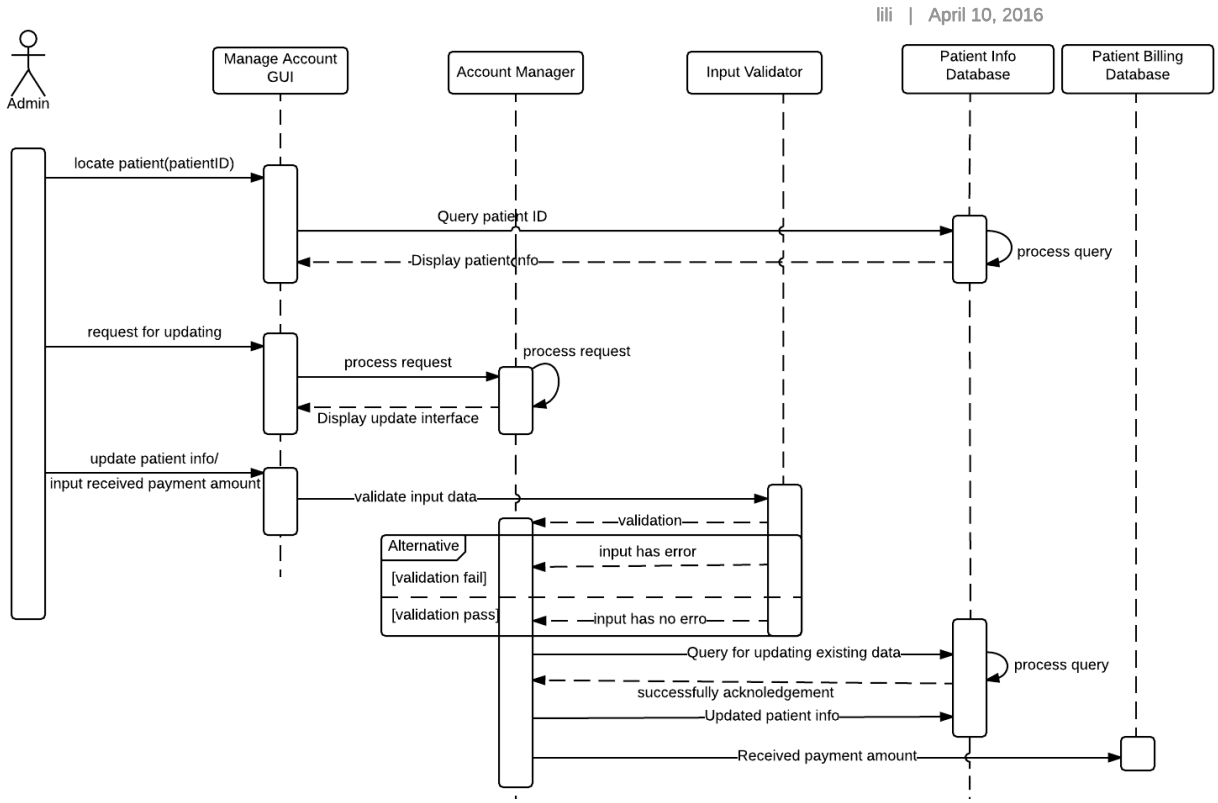
Figure 4


lili | April 10, 2016
Admin
Patient Claim GUI
Patient Info Database
HCFA Form
Input Validator
Patient Billing Database
Insurance Company
locate patient (PatientID)
Query for patient(patientID)
acknowledgement
process query
patient demographics and invoice
Request to fill up HCFA Form
process request
Dispaly HCFA Form
Fills HCFA Form
Validate input
Validation
Alternative
[validation pass]
input is valid
[validation fail]
input is invalid
Filled HCFA Form
HCFA Form

1.1.5 Module 5 Detail - Manage Patient Account

Use Case Name: Manage Patient Account	ID: UC05
Primary Actor: Admin	
Description: This use case describes how an admin manages a patients account	
Trigger: The user receives updated information for the patient, and/or payment from either the patient, or insurance company	
Precondition: The user must be authorized, and the claim has been approved by the insurance company	
Post condition: The patients account in Patient Account has been updated successfully	
Basic Flow: 1. The actor searches for the patient using their patient ID. 2. The system returns the search results by displaying the patients account page. 3. The actor updates the patient's information, including payments received from either the insurance company, patient, or both. 4. The updated patient's info is stored in Patient Info Database, and the received payment information is stored in the Patient Billing Database. 5. The use case ends.	
Alternative Flows: *1. The actor enters invalid information in the claim form: If the system determines that the user entered invalid information, the following occurs:* 1) The system displays a message when a field is required. 2) The system prompts the User to re-enter the invalid information. 3) The user re-enters the information, and the system re-validates it. 4) If valid data is entered, the updated information is stored. 5) If invalid data is entered, the Invalid alternative flow is executed again. This continues until the User enters valid information, or chooses Cancel. 6) Invalid Patient Account information includes: - Missing information items - The input data that is entered does not comply to its definition in the glossary - Improperly formed information, such as an email address	

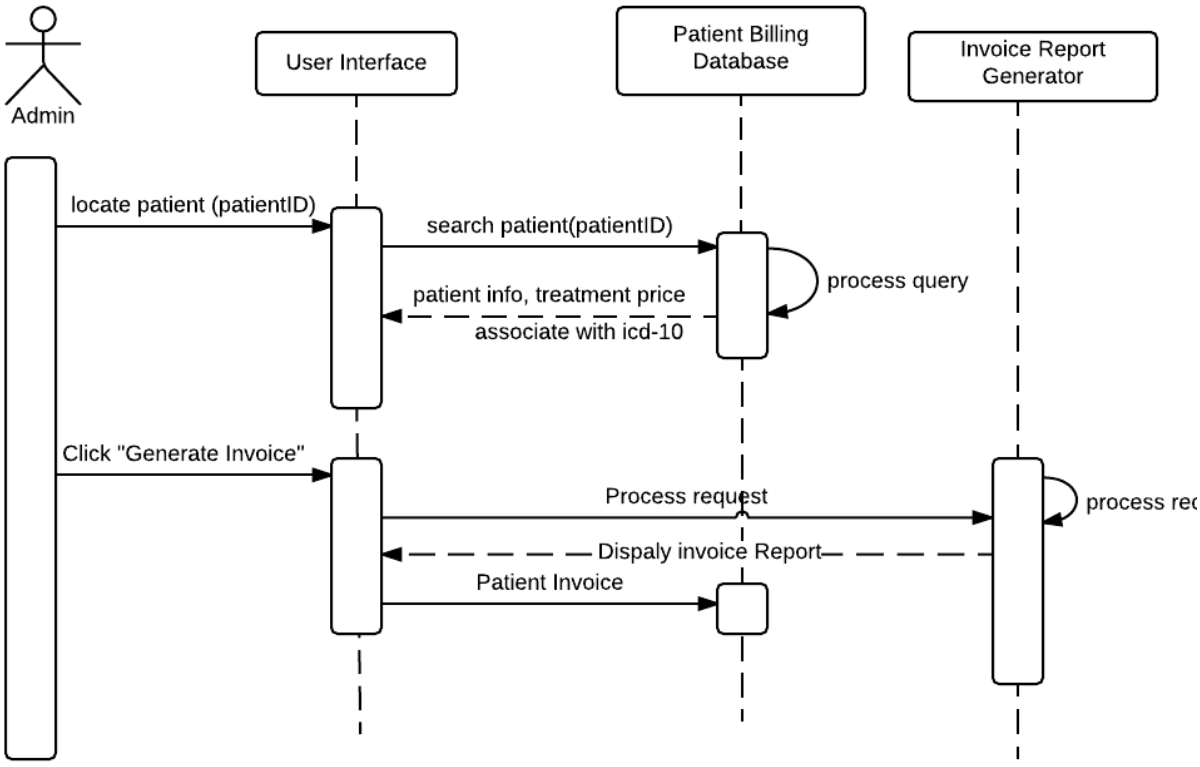
Figure 5


lili | April 10, 2016
Admin
Manage Account GUI
Account Manager
Input Validator
Patient Info Database
Patient Billing Database
locate patient(patientID)
Query patient ID
process query
Display patient info
request for updating
process request
process request
Display update interface
update patient info/
input received payment amount
validate input data
validation
Alternative
input has error
[validation fail]
[validation pass]
input has no erro
Query for updating existing data
process query
successfully acknoledgement
Updated patient info
Received payment amount

1.1.6 Module 6 Detail - Produce Invoice

Use Case Name: Produce Invoice	ID: UC6
Primary Actor: Admin	
Description: This use case describes how an admin produces an invoice	
Trigger: The Admin clicks the “Produce Invoice” button	
Precondition: The doctor has input the ICD-10 code	
Post condition: Invoices have been produced and stored	
Basic Flow: 1. The actor searches for the patient using their patient ID. 2. The system displays the patient’s information, including their ICD-10 codes, which are entered by patient’s doctor. 3. The actor clicks “generate invoice” button. 4. The new invoice is stored in the Patient Billing data store. 5. The use case ends.	
Alternative Flows:	

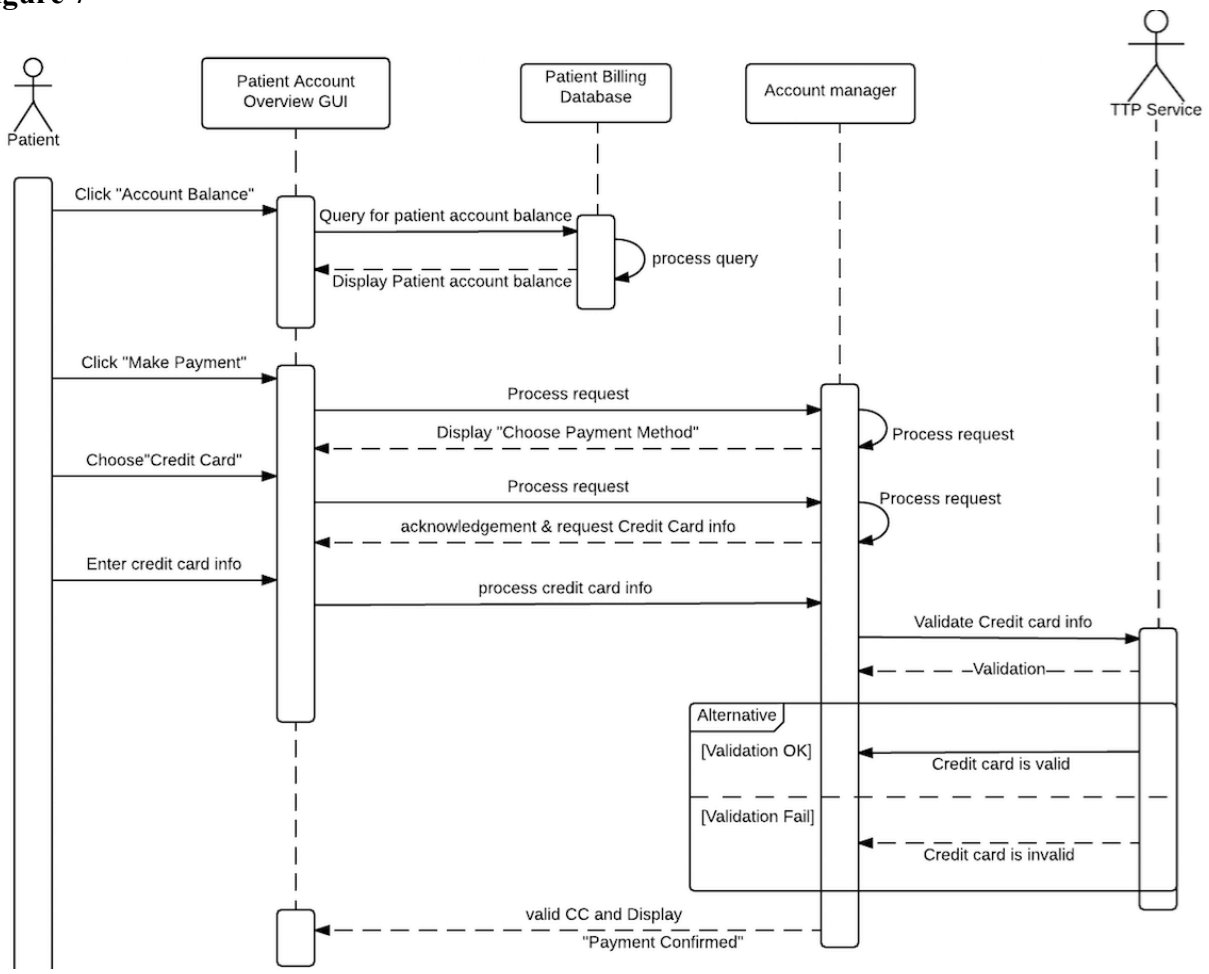
Figure 6


Admin
User Interface
Patient Billing Database
Invoice Report Generator
locate patient (patientID)
search patient(patientID)
process query
patient info, treatment price
associate with icd-10
Click "Generate Invoice"
Process request
Dispaly invoice Report
Patient Invoice

1.1.7 Module 7 Detail - Make Payment

Use Case Name: Make a Payment	ID: UC07
Primary Actor: Patient	
Description: This use case describes how a patient makes a payment.	
Trigger: The user saw the balance due in his/her account	
Precondition: The user must be authorized, and has bills that need to be paid	
Post condition: The system display "Payment Confirmed"	
Basic Flow: 1. The actor clicks "Account Balance" from Account Overview. 2. The system displays the patient account balance. 3. The actor chooses to make a payment if his/her account balance has not been paid off. 4. The actor chooses pay by credit card, cash, or check. 5. If the actor pays by credit card, he/she has to enter the credit card information. 6. The Payment will then be charged by the TTP (Trusted Third Party) Service. 7. The "Payment Confirmed" message will be displayed if the credit card has been approved. 8. The use case ends.	
Alternative Flows: 1. Payment doesn't confirm: If the TTP Service doesn't approve the credit card, the payment will not be confirmed by the system.	

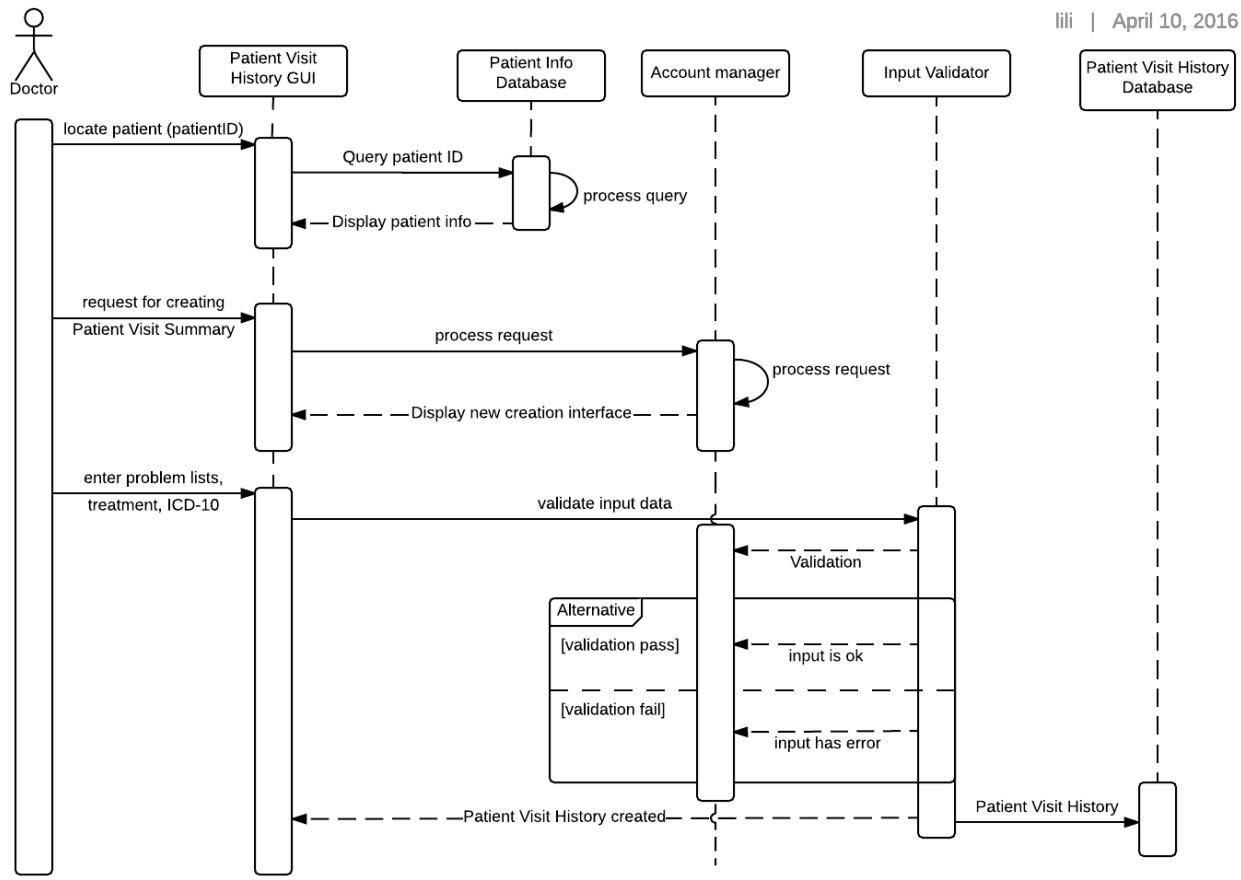
Figure 7


Patient
Patient Account Overview GUI
Patient Billing Database
Account manager
TTP Service
Click "Account Balance"
Query for patient account balance
process query
Display Patient account balance
Click "Make Payment"
Process request
Display "Choose Payment Method"
Process request
Choose"Credit Card"
Process request
Process request
acknowledgement & request Credit Card info
Enter credit card info
process credit card info
Validate Credit card info
Validation
Alternative
[Validation OK]
Credit card is valid
[Validation Fail]
Credit card is invalid
valid CC and Display
"Payment Confirmed"

1.1.8 Module 8 Detail - Produce Patient Visit Summary

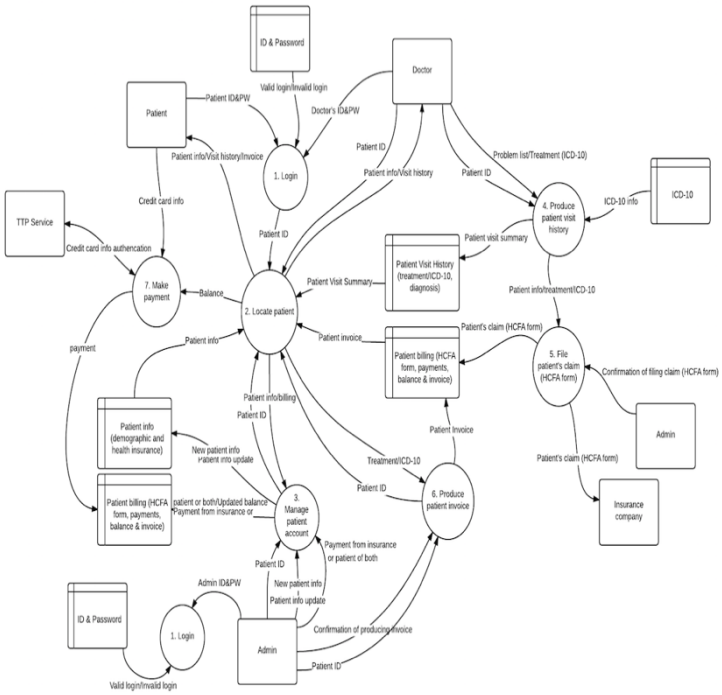
Use Case Name: Produce a Patients Visit Summary	ID: UC08
Primary Actor: Doctor	
Description: This use case describes how a doctor produces his/her Patients Visit Summary in the system	
Trigger: The patient has visited the doctor	
Precondition: The user must be authorized, and have seen a patient	
Post condition: A Patient Visit Summary has been produced and stored in the Patient Visit Summary data store	
Basic Flow: 1. The actor enters the patient's ID into the search engine. 2. The system displays the Patients Information. 3. The actor inputs the Patient Visit Summary, including the Patient Problem List, Treatment, and ICD-10 code. 4. The Patient Visit Summary is stored in Patient Visit Summary data store. 5. The use case ends.	
Alternative Flows: *1. The actor enters invalid information in the claim form: If the system determines that the user entered invalid information, the following occurs:* 1) The system displays a message when a field is required. 2) The system prompts the User to re-enter the invalid information. 3) The user re-enters the information and the system re-validates it. 4) If valid information is entered, the Patient Visit Summary is stored. 5) If invalid information is entered, the Entered Information is Invalid alternative flow is executed again. This continues until the User enters valid information, or chooses Cancel. 6) Invalid Patient Account information include: - Missing information items - Claim form information entered does not comply to its definition in the glossary - Not well formed information	

Figure 8


lili | April 10, 2016
Doctor
Patient Visit History GUI
Patient Info Database
Account manager
Input Validator
Patient Visit History Database
locate patient (patientID)
Query patient ID
process query
Display patient info
request for creating Patient Visit Summary
process request
process request
Display new creation interface
enter problem lists, treatment, ICD-10
validate input data
Validation
Alternative
[validation pass]
input is ok
[validation fail]
input has error
Patient Visit History
Patient Visit History created

1.2 Data Flow Diagram

Figure 9

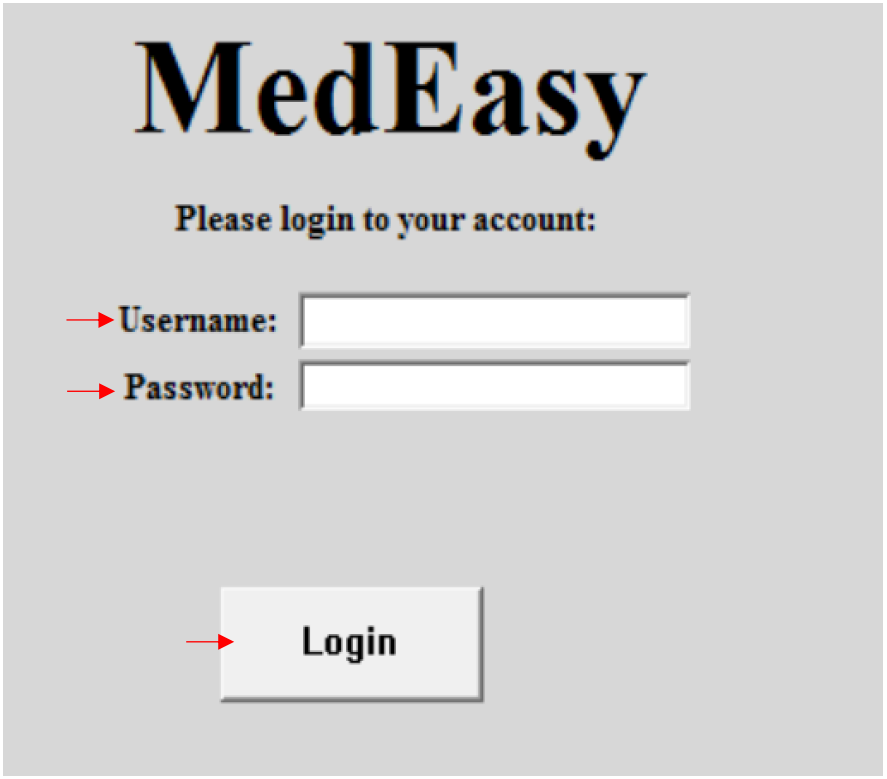


2. User Manual

2.1 Patient

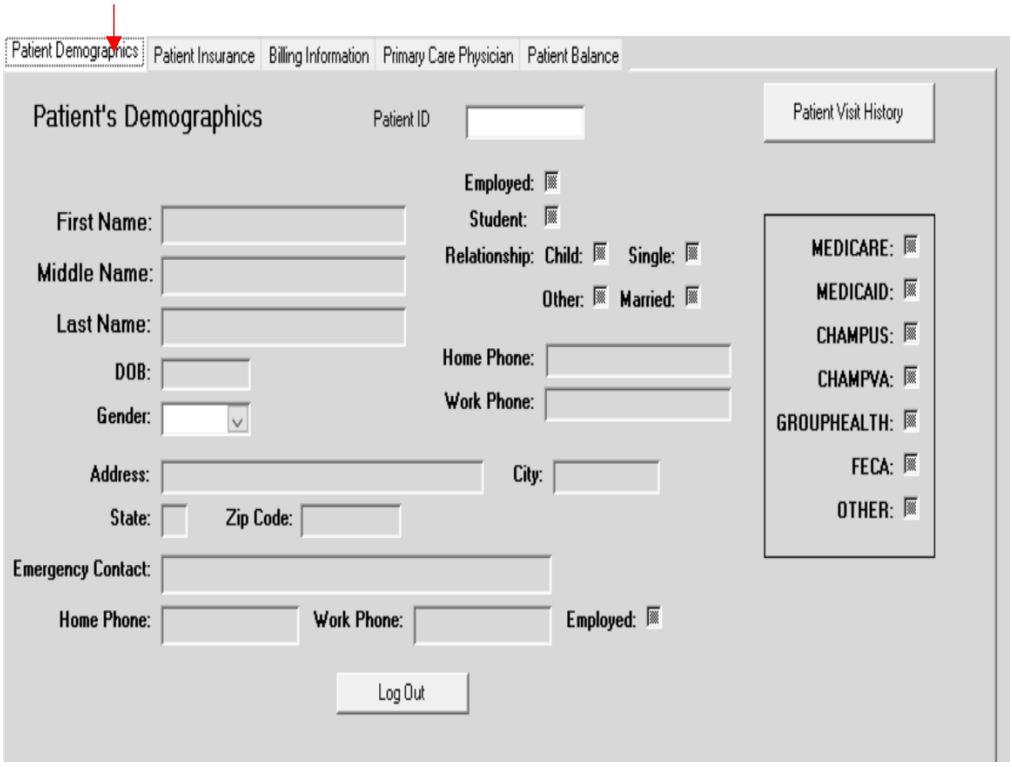
2.1.1 Logging into the System

Any user that has been granted access into the MedEasy system will be provided their login information by the administrator. All users will first access the login portal below, but as a “Patient”, you will input your provided “Username” and “Password” Once the prior criteria has been provided, you will then click “Login”. **(GUI01-1)**



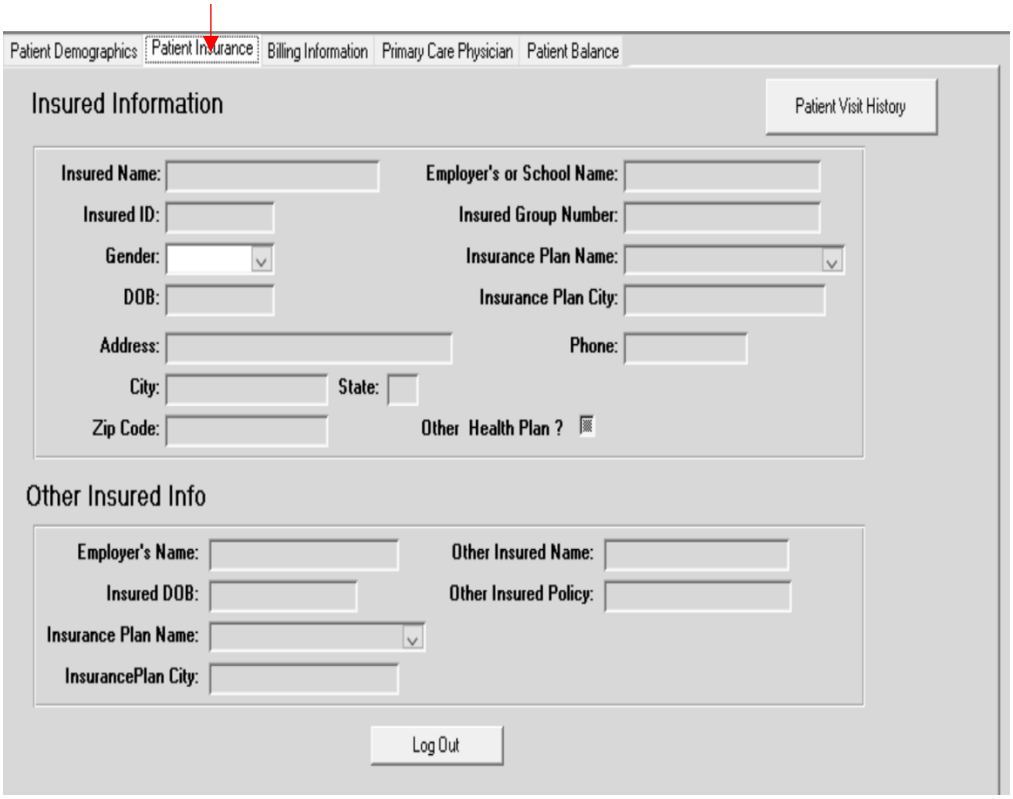
2.1.2 Patient Demographics

Once you have logged into the system, you will click on the information you would like to change, and you will have the ability to correct or modify the fields as you see fit. The information is automatically saved when entered, so no extra work is required to save. You may Log out of the system at any time, by clicking the "Log Out" button located at the bottom of the screen. **(GUI02-1)**



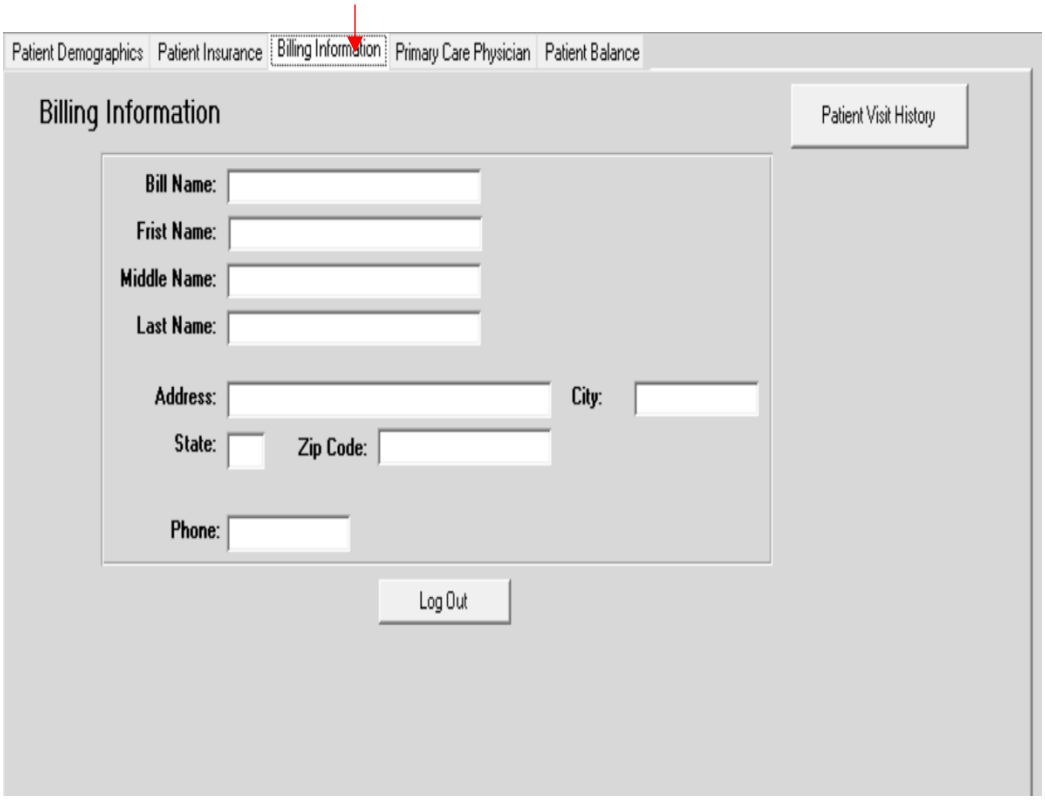
2.1.3 Insurance

You may change any information that has changed on this screen, and you may log out of the system at any time by clicking the “Log Out” button, located at the bottom of the screen. **(GUI03-1)**



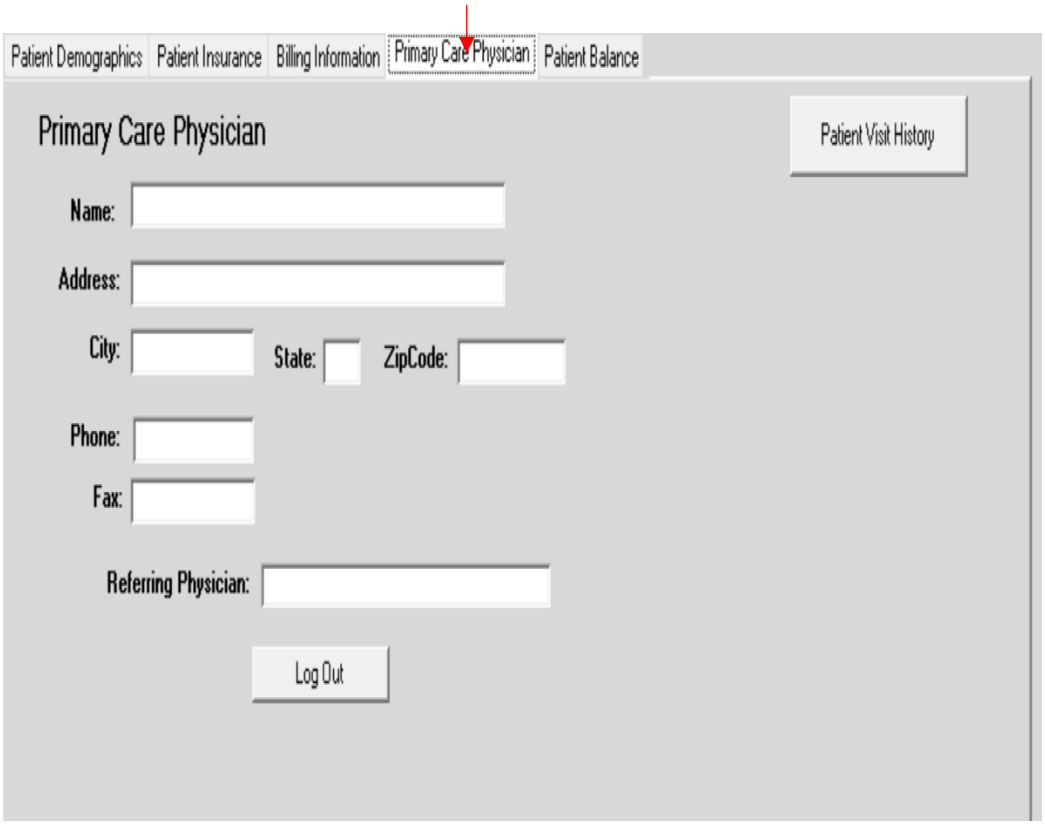
2.1.4 Billing

You may change any information that has changed on this screen, and you may log out of the system at any time by clicking the “Log Out” button, located at the bottom of the screen. **(GUI04-1)**



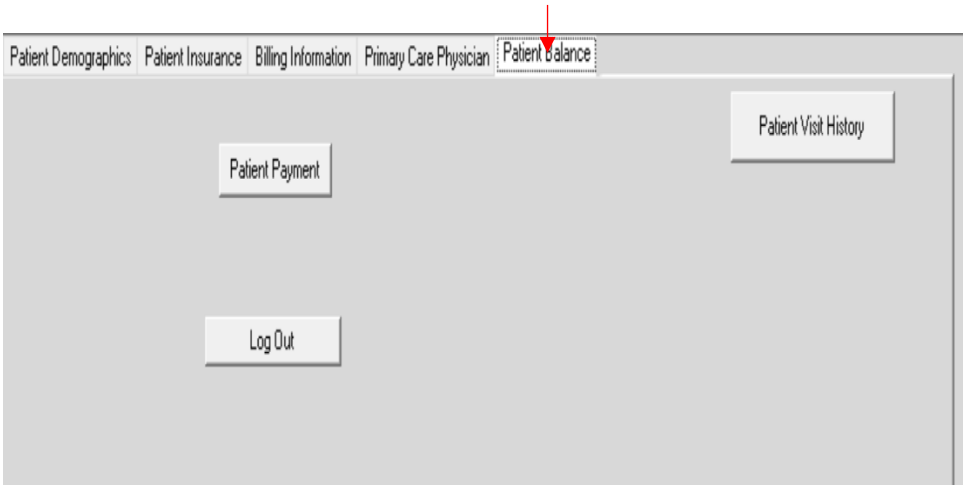
2.1.5 Primary Physician

You may change any information that has changed on this screen, and you may log out of the system at any time by clicking the “Log Out” button, located at the bottom of the screen. **(GUI05-1)**



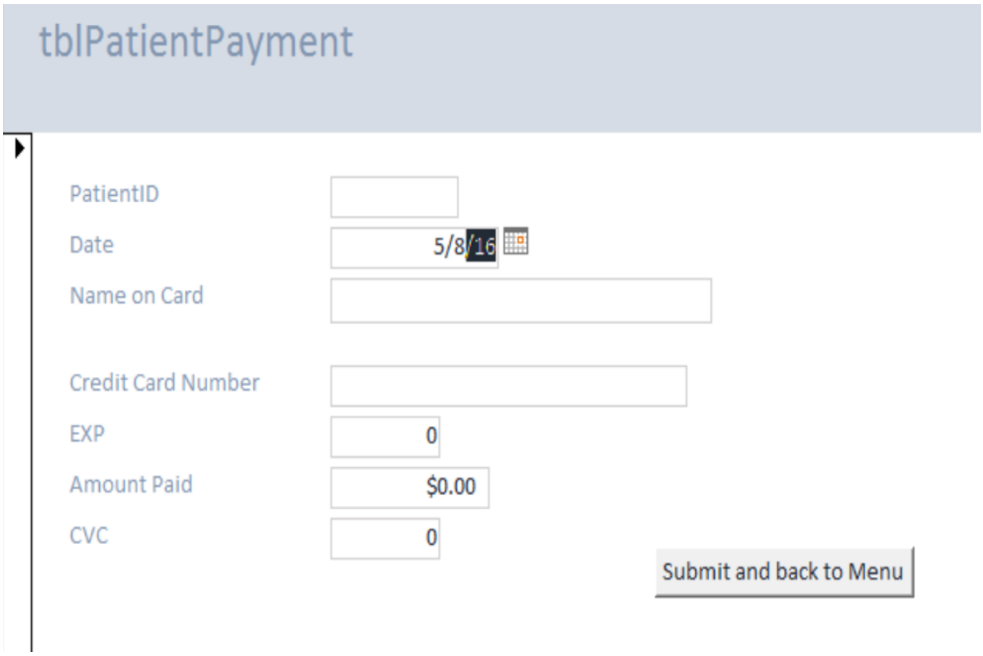
2.1.6 Patient Balance

You may Log out of the system at any time, by clicking the “Log Out” button located at the bottom of the screen. I wasn’t able to get a more up to date screenshot, if this made it in. **(GUI06-1)**



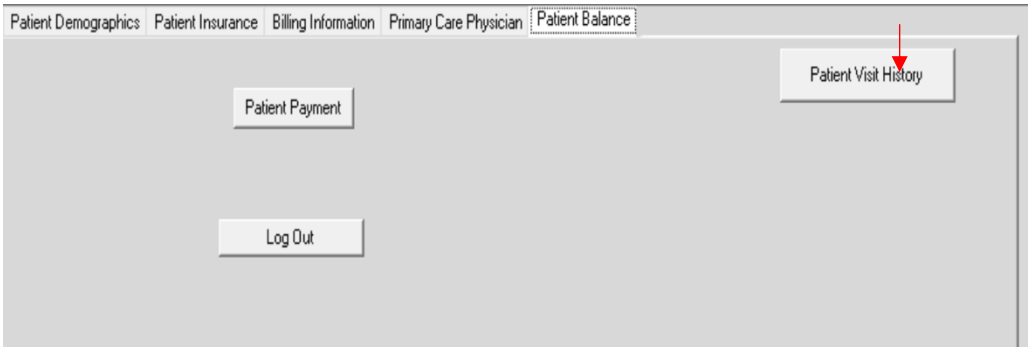
2.1.6.1 Payment

After entering the amount, you would like to pay, and your payment information, you will then click “Submit and back to Menu” to be brought back to the previous screen. **(GUI07-1)**



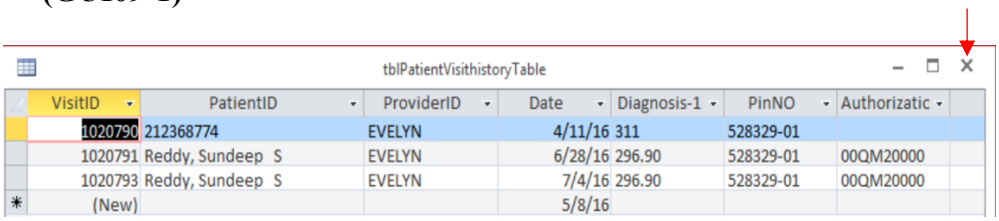
2.1.7 Patient History

You may Log out of the system at any time, by clicking the “Log Out” button located at the bottom of the screen. When clicked, a display box will appear, so that a report may be generated. **(GUI08-1)**



2.1.7.1 Visit History

When the user is finished viewing their visit history, they will click the “x” in the upper right corner of the screen to go back to the last screen. **(GUI09-1)**



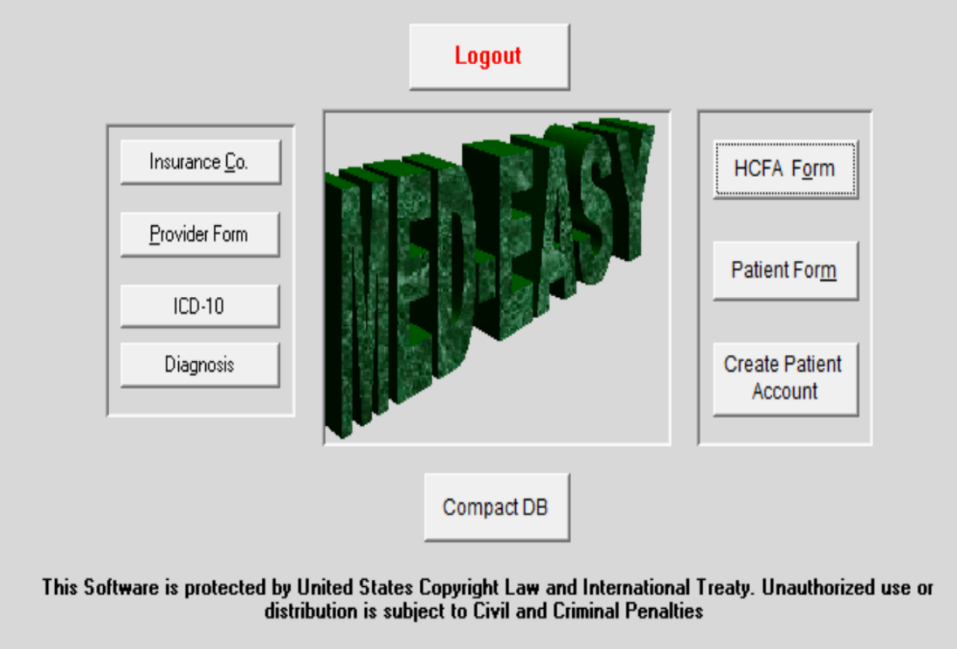
2.2 Doctor

2.2.1 Logging into the System

Any user that has been granted access into the MedEasy system will be provided their login information by the Administrator. All users will first access the login portal below, but as a “Doctor”, you will input your provided “Username” and “Password. Once the prior criteria has been provided, you will then click “Login”. **(GUI01-2)**

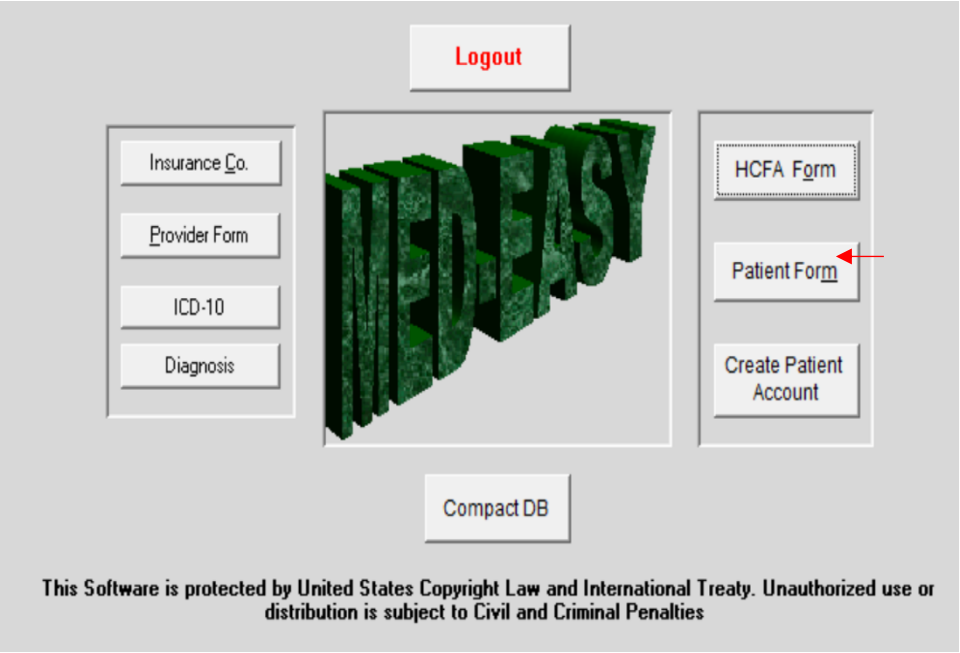
2.2.2 Main Menu

After logging into the system, you will be presented with the “Main Menu” of the system. This will have two options, one where you can access forms, such as “HCFA”, and another where you can input information. You will also have the option to “Log Out” of the system, which is located at the top of the “Main Menu” page. **(GUI02-2)**



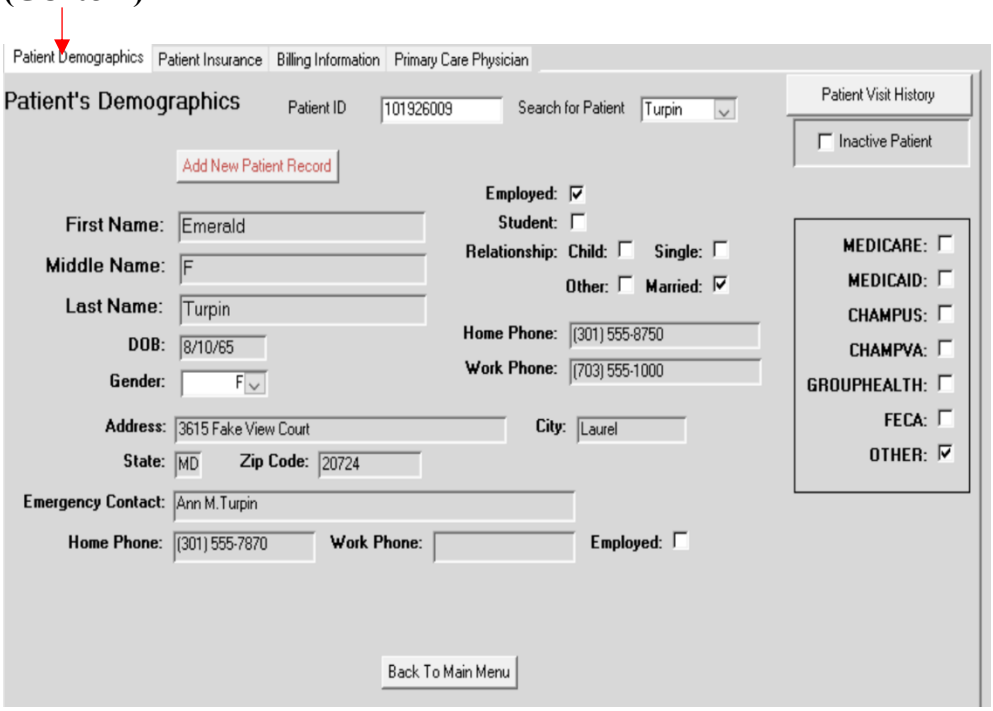
2.2.3 Patient Form

When selecting the “Patient Form” button, you will be directed to the “Patient Demographics” menu. In addition, you will be able to navigate to the patients “Insurance”, “Billing”, “Primary Physician”, and “Patient History” sections when desired. When you wish to move between pages, simply click the appropriate button located on the left hand side of the screen. When located on editable pages, you will click the information you would like to change, and you will have the ability to modify the fields as you see fit. The information is automatically saved when entered, so there is no extra work that is required to save the information. **(GUI02-2-1)**



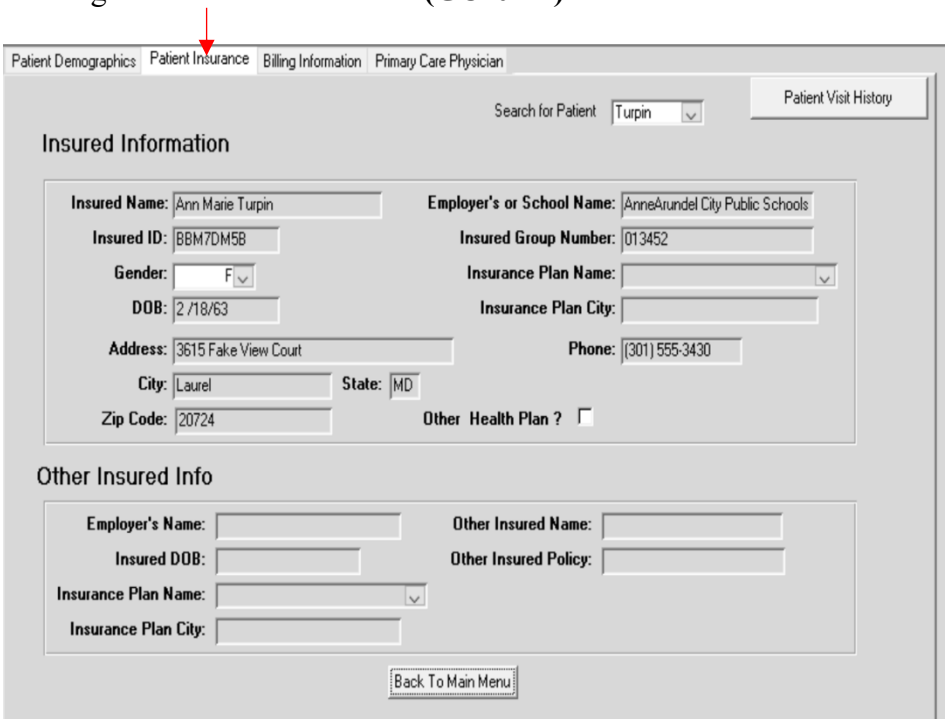
2.2.3.1 Patient Demographics

When located on this page, you will click the information you would like to change, and you will have the ability to correct or modify the given fields as you see fit. You may go back to the main menu at any time, simply by clicking “Back to Main Menu”. **(GUI03-2)**



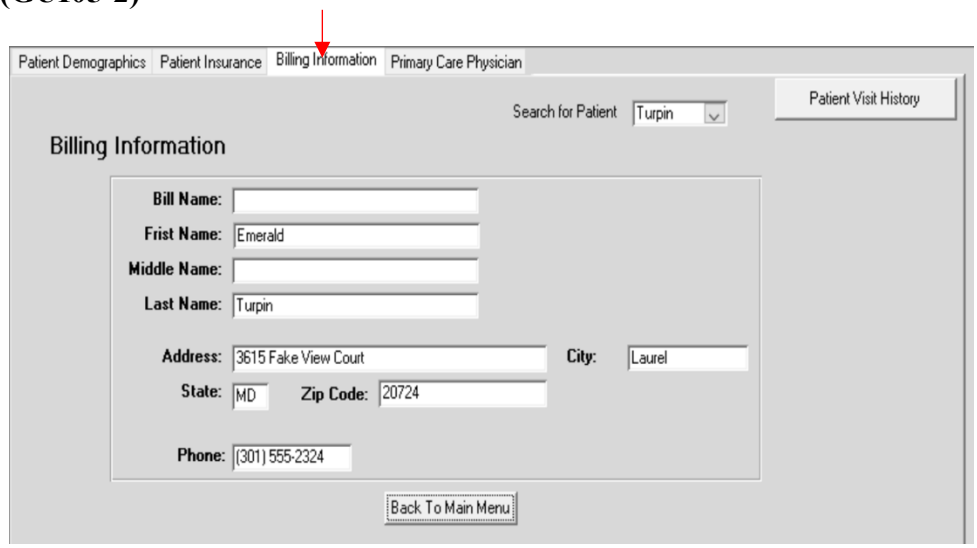
2.2.3.2 Insurance

Any changes can be made to the information on this page when needed. You may go back to the main menu at any time, simply by clicking “Back to Main Menu”. **(GUI04-2)**



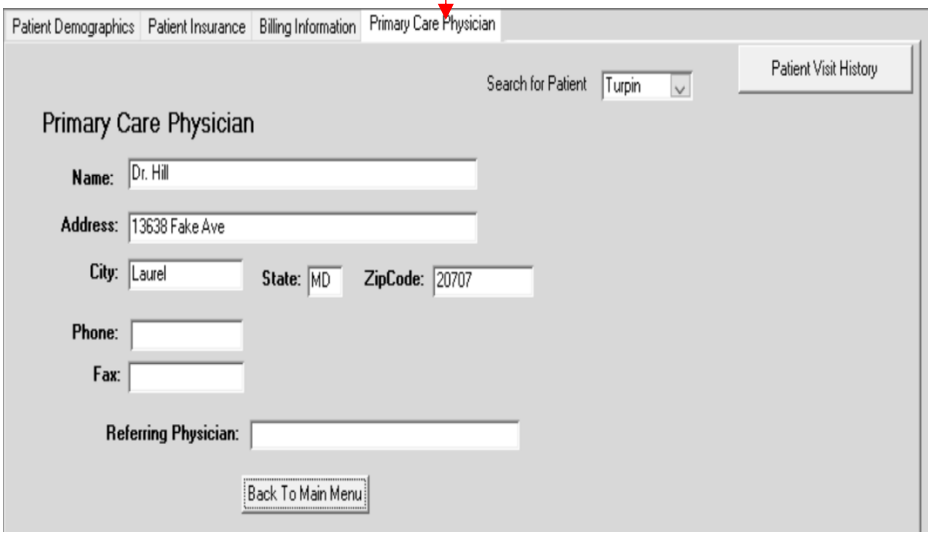
2.2.3.3 Billing

All fields are editable on this screen, and ou may go back to the main menu at any time, simply by clicking “Back to Main Menu”. **(GUI05-2)**



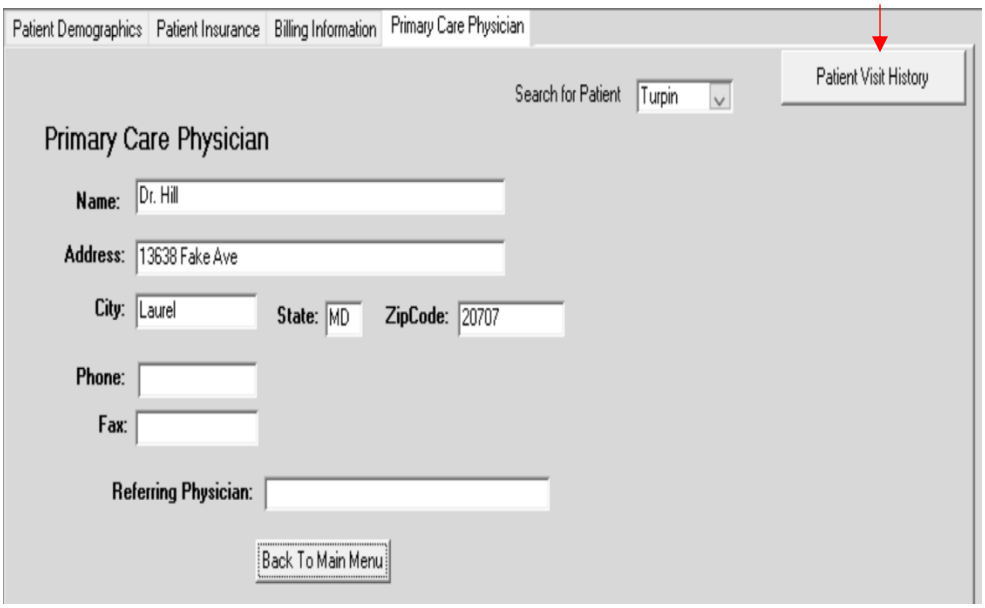
2.2.3.4 Primary Physician

You will have the ability to change all of the information on the page, and you may go back to the main menu at any time, simply by clicking “Back to Main Menu”. **(GUI06-2)**



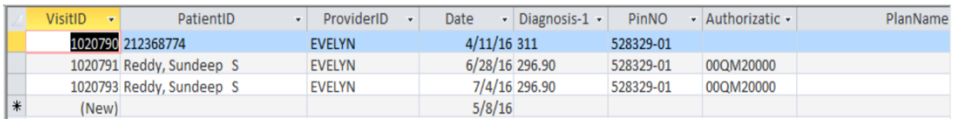
2.2.3.5 Patient History

You may Log out of the system at any time, by clicking the "Log Out" button located at the bottom of the screen. When clicked, a display box will appear, so that a report may be generated. **(GUI07-2)**



2.2.3.5.1 Patient Visit History

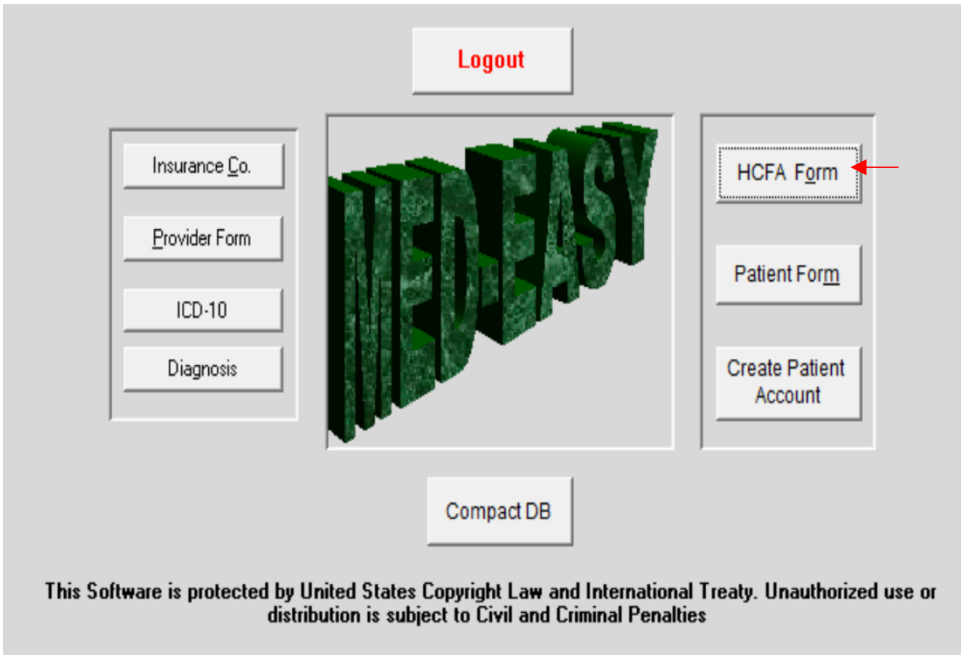
When the user is finished viewing their visit history, they will click the "x" in the upper right corner of the screen to go back to the last screen. **(GUI08-2)**



VisitID	PatientID	ProviderID	Date	Diagnosis-1	PinNO	Authorizatic	PlanName
1020790	212368774	EVELYN	4/11/16	311	528329-01		
1020791	Reddy, Sundeep S	EVELYN	6/28/16	296.90	528329-01	00QM20000	
1020793	Reddy, Sundeep S	EVELYN	7/4/16	296.90	528329-01	00QM20000	
(New)			5/8/16				

2.2.4 HCFA

When selecting the “HCFA” button, you will be directed to the “HCFA” form, where you will have access to input all of the various information that is required within the form. **(GUI02-2-2)**



2.2.4.1 Navigation

Please click the desired button when entering the information. **(GUI09-2)**

2.2.4.1.1 Top Panel

2.2.4.1.1.1 Lookup Record

This option provides a search bar that will allow you to pull the HCFA forms for a specified patient. **(GUI010-2)**

2.2.4.1.1.2 Show All Records

This option provides all records for the selected individual from the “Lookup Visit/Patient”. **(GUI11-2)**

2.2.4.1.1.3 Add Record

This option generates a new HCFA Form when selected for an individual. **(GUI12-2)**

2.2.4.1.1.4 Delete Record

This option will remove the HCFA form from the system. **(GUI13-2)**

2.2.4.1.1.5 Add Patient

This option provides the ability to add a new patient so a HCFA Form can be created. **(GUI14-2)**

2.2.4.1.1.6 Menu

This option will take you back to the forms menu. **(GUI15-2)**

2.2.4.1.2 Entering Information

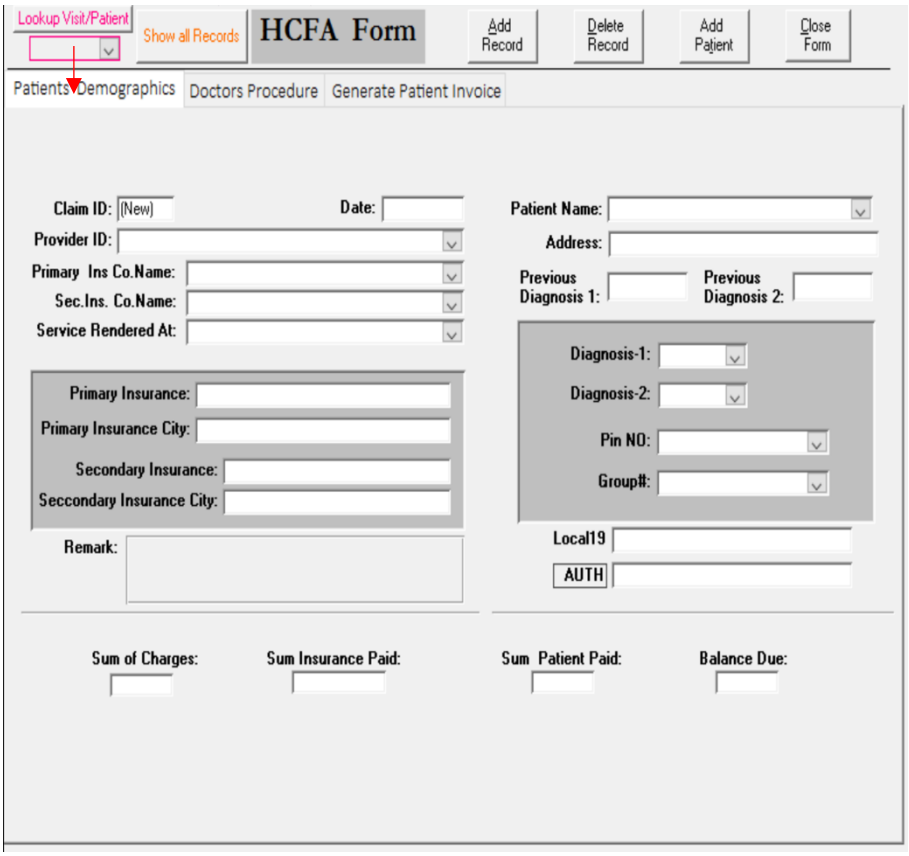
Once you have selected the correct patient, and have their records pulled, please enter the appropriate information into the various fields. **(GUI16-2)**

2.2.4.1.3 Entry Options

You may put information into three different areas, “Patient Demographics”, “Doctors Procedure”, or “Generate Patient Invoice”.

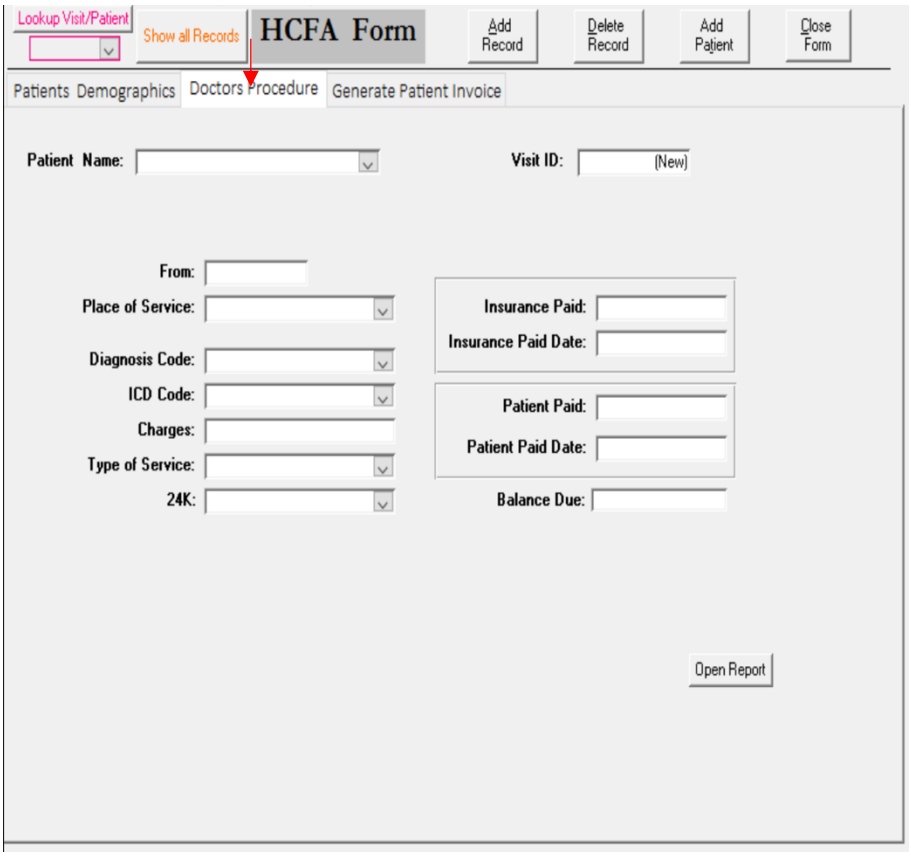
2.2.4.1.3.1 Patient Demographics

(GUI09-2-1)



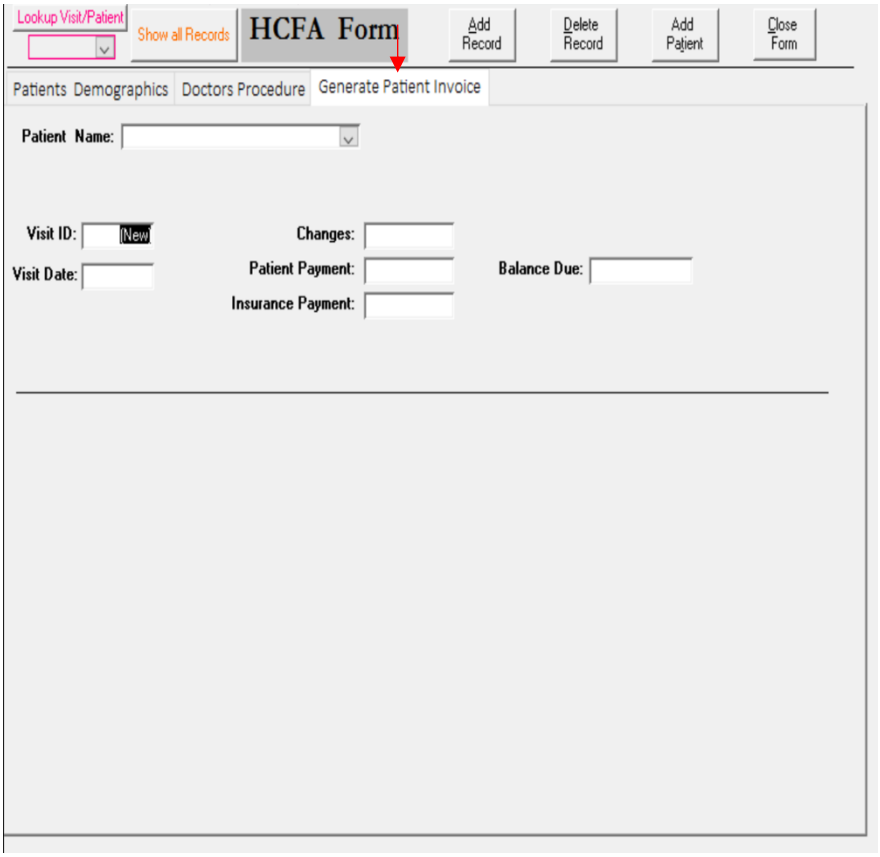
2.2.4.1.3.2 Doctors Procedure

This is where the doctor will enter the information from the previous visit. **(GUI17-2)**



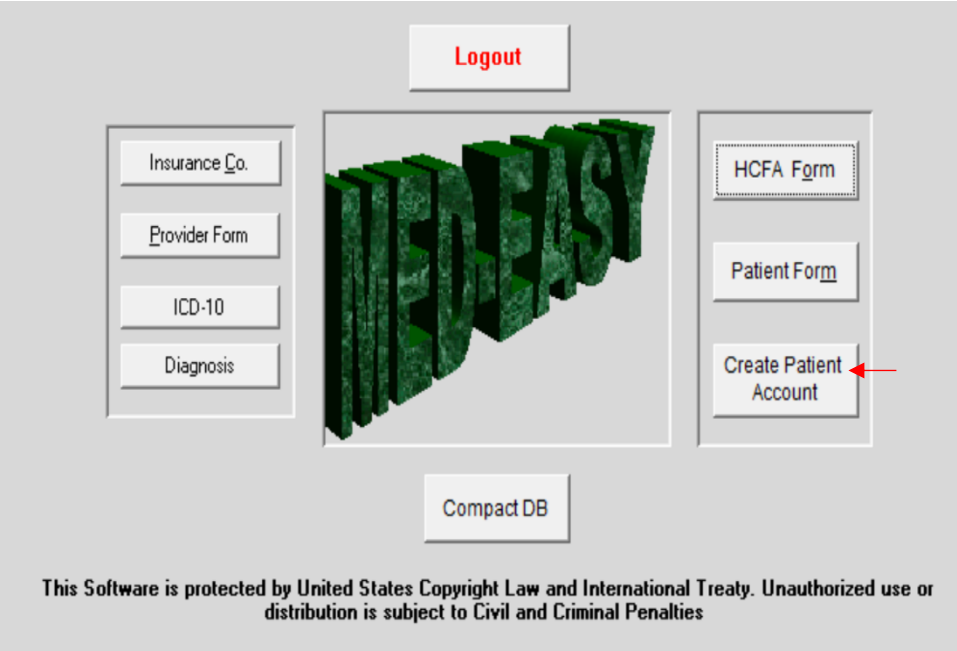
2.2.4.1.3.3 Generate Patient Invoice

This is the section of the program that creates the invoice for the various patients. **(GUI18-2)**



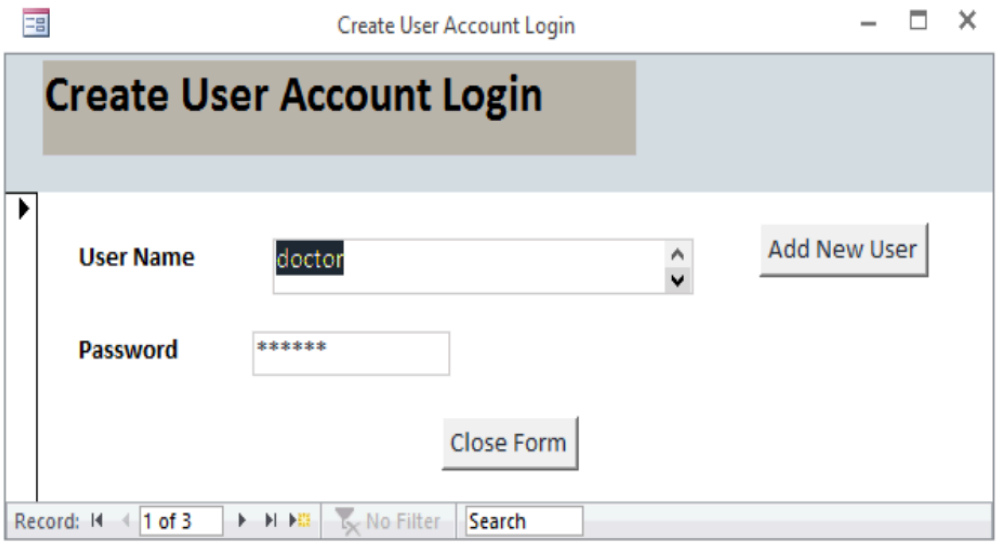
2.2.5 Create Patient Account

From the “Main Menu” screen, you will select “Create Patient Account”. **(GUI02-2-3)**



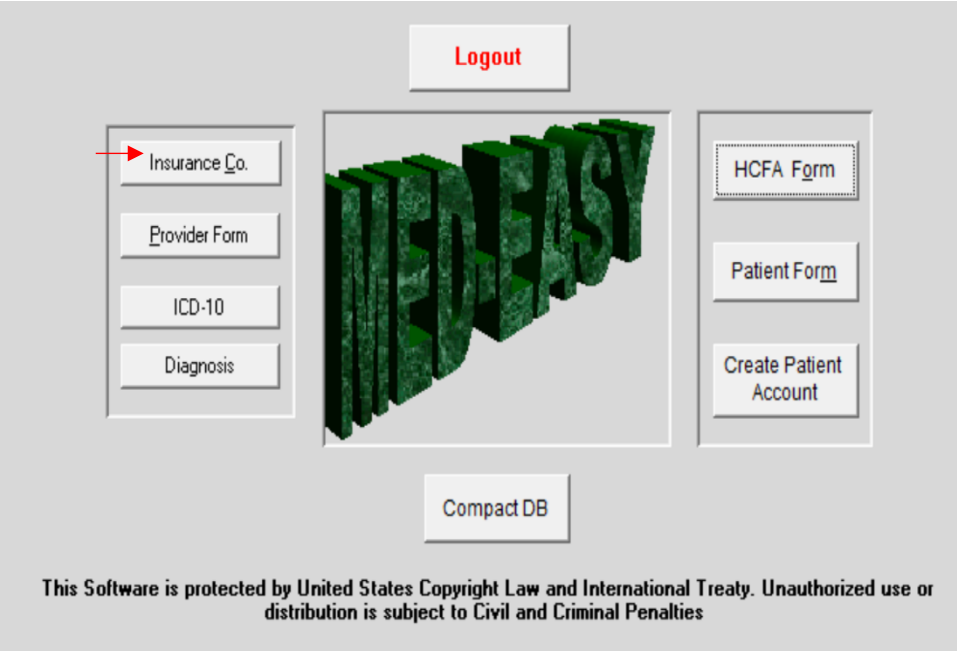
2.2.5.1 Data Entry

When on this page, the user will enter new users “username”, and “password”, and once completed, they will click “Add New User”. Once finished, the user may click “Close Form” to go back to the main menu. **(GUI19-2)**



2.2.6 Insurance Company

From the “Main Menu” screen, you will select “Insurance Co.”. **(GUI02-2-4)**

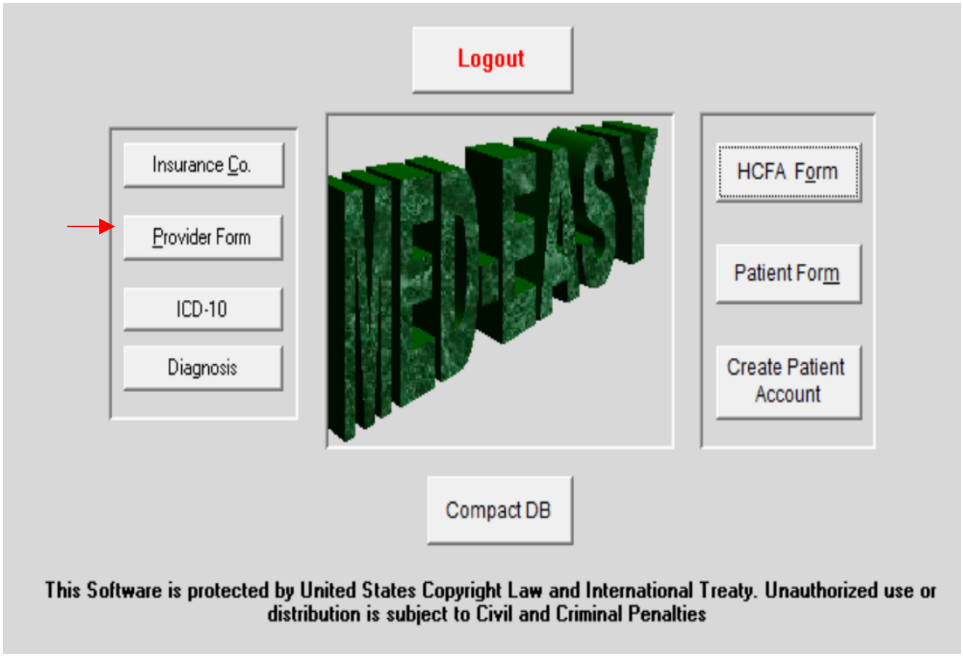


2.2.6.1 Data Entry

After selecting “Insurance Co.”, you will be brought to the screen below, where you will enter your information, and click “Add Record”. You will have the option to return to the main menu of the system, by clicking the button on the left hand side of the screen, named “Back to Main Menu”. **(GUI20-2)**

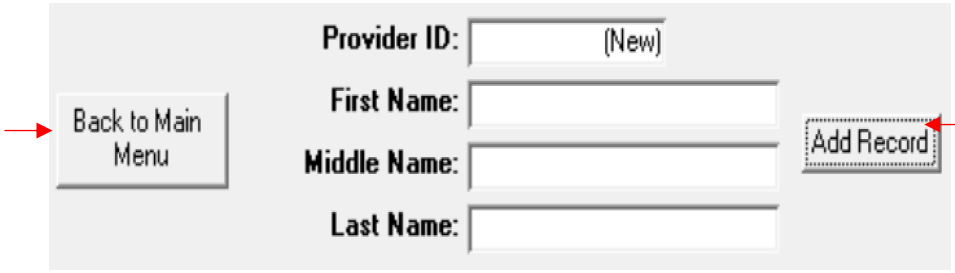
2.2.7 Provider Form

From the “Main Menu” screen, you will select “Provider Form”. **(GUI02-2-5)**



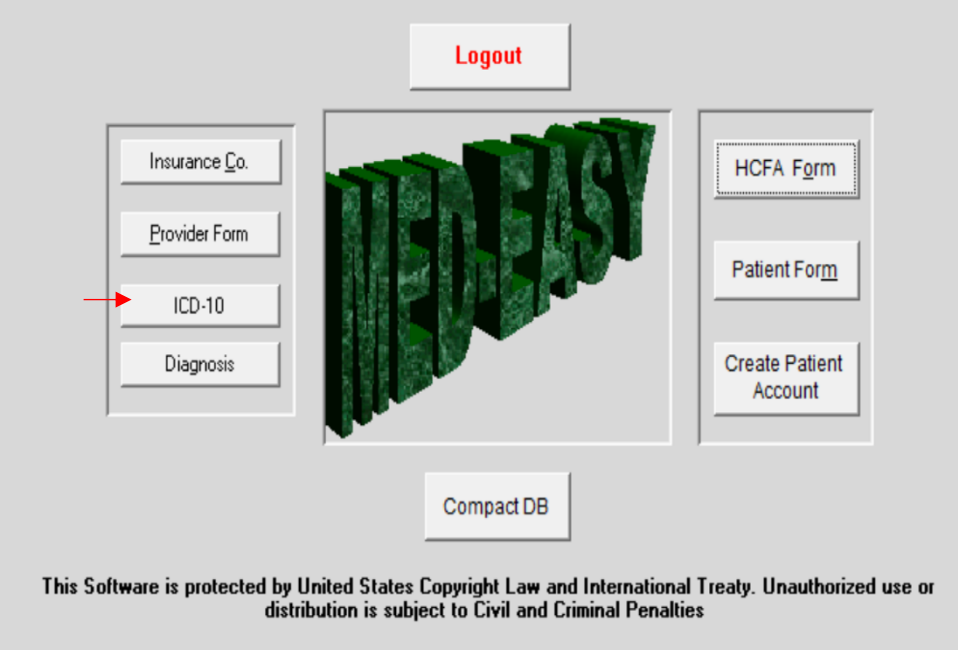
2.2.7.1 Data Entry

When you want to add a new provider into the system, you need to add the required information into the textboxes, and then click “Add Record” button, in order to save the new information into the system. You will them click “Back to Main Menu”, to go back to the main menu. **(GUI21-2)**



2.2.8 ICD-10

From the “Main Menu” screen, you will select “ICD-10”. **(GUI02-2-6)**

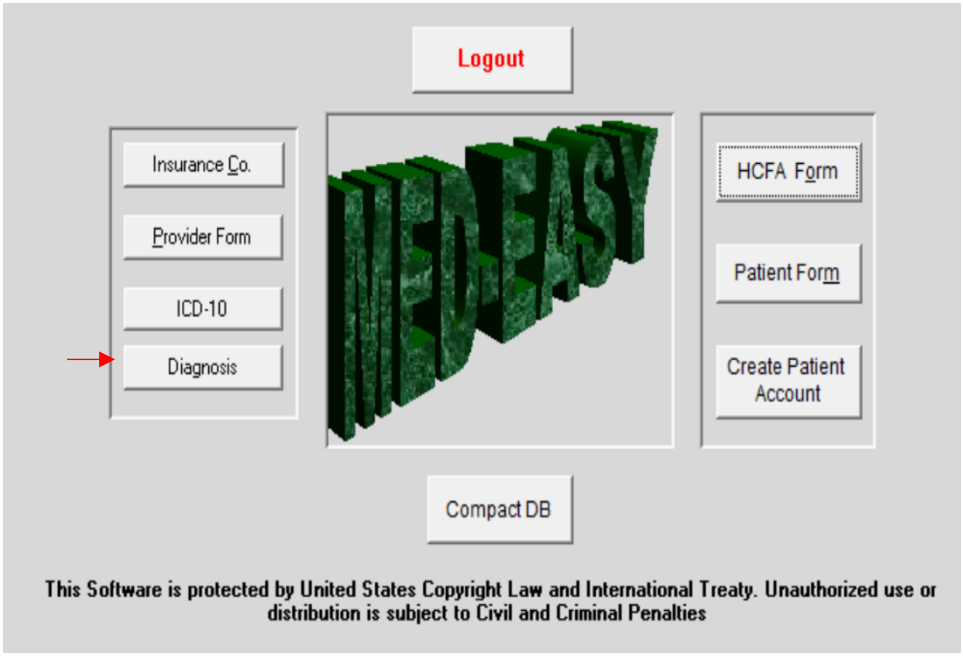


2.2.8.1 Adding Data

When adding a new ICD-10 code into the system, you will enter the information into the various textboxes, and click the “Add Record” button to save the new information into the system. You may then press the “Back to Main Menu”, to go back to the main menu. **(GUI22-2)**

2.2.9 Diagnosis

From the “Main Menu” screen, you will select “Diagnosis”. **(GUI02-2-7)**



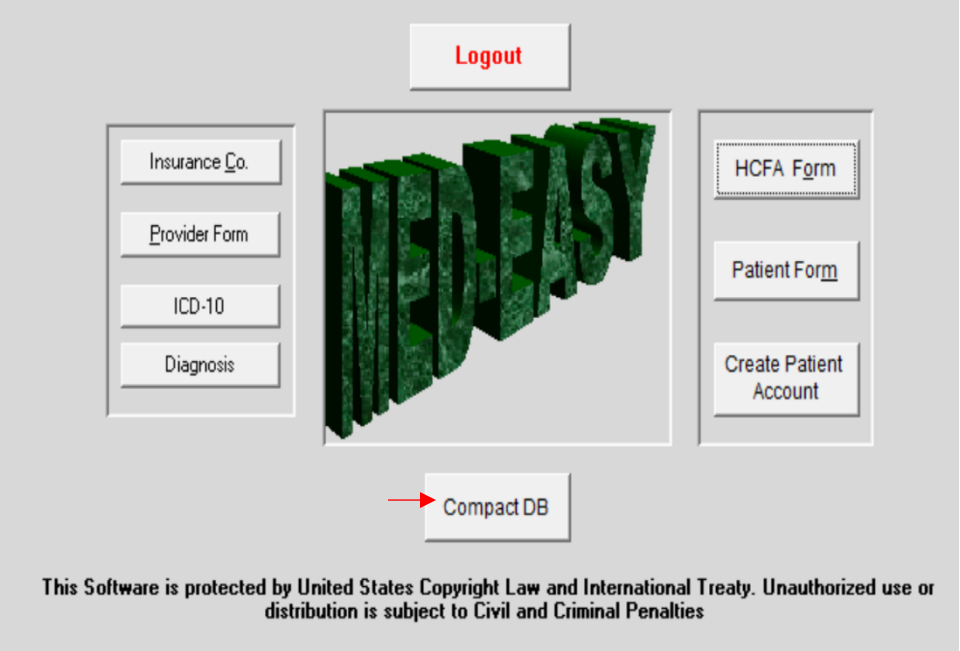
2.2.9.1 Data Entry

When you want to add new information into the system, you will enter the information into the various text boxes, and then click “Add Record” button to save the new information into the system. You may then press the “Back to Main Menu”, to go back to the main menu. **(GUI23-2)**

2.2.10 Compact Database

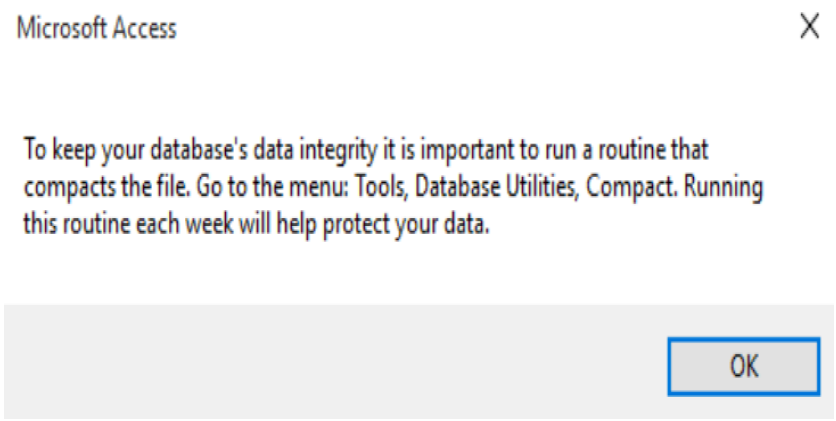
From the “Main Menu” screen, you will select “Compact Database”.

(GUI-2-2-8)



2.2.10.1 Display

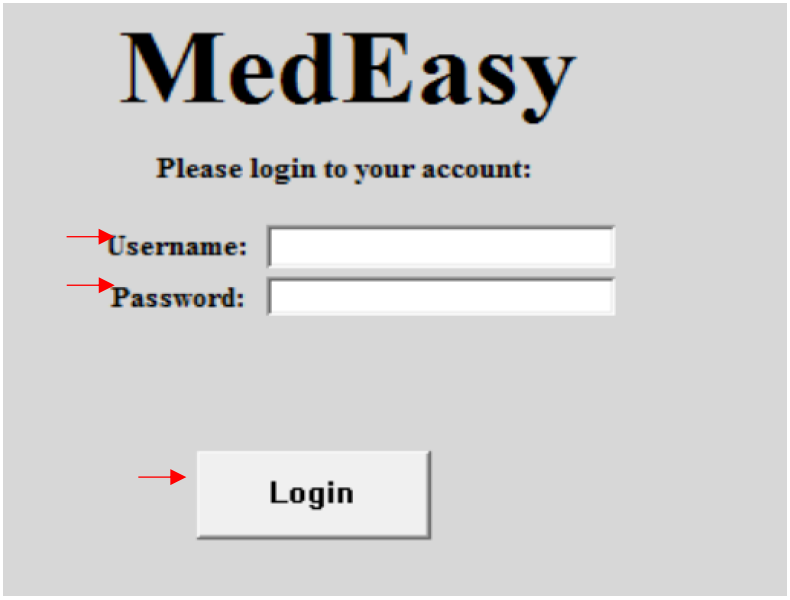
This button will only display a message to remind the primary users of the system to occasionally compact their database, and the necessary steps to accomplish it. **(GUI24-2)**



2.3 Administrator

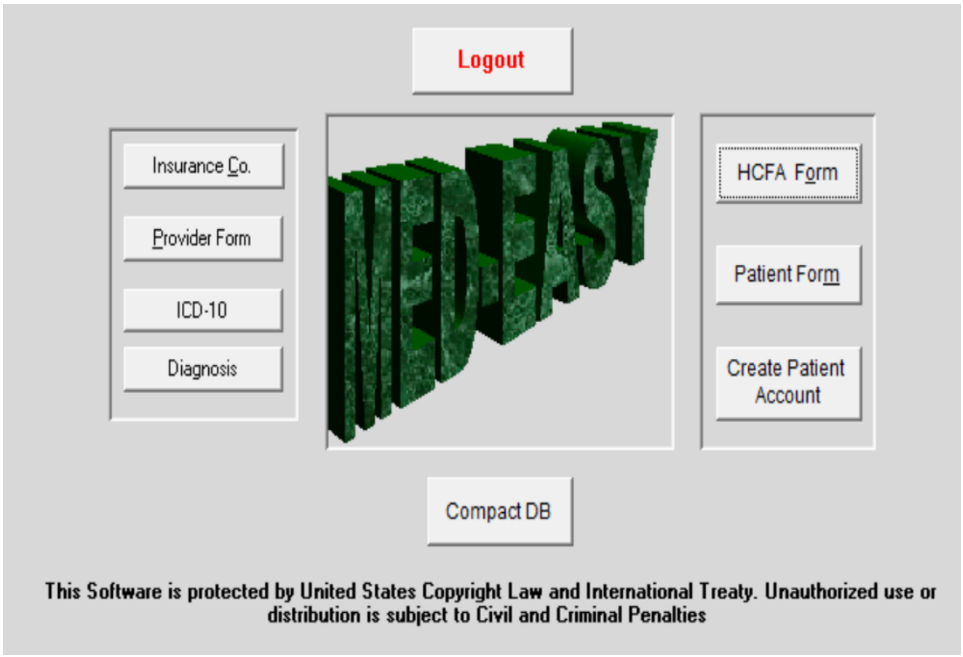
2.3.1 Logging into the System

Any user that has been granted access into the MedEasy system will be provided their login information by the Administrator. All users will first access the login portal below, but as a “Administrator”, you will input your provided “Username” and “Password. Once the prior criteria has been provided, you will then click “Login”. **(GUI01-3)**



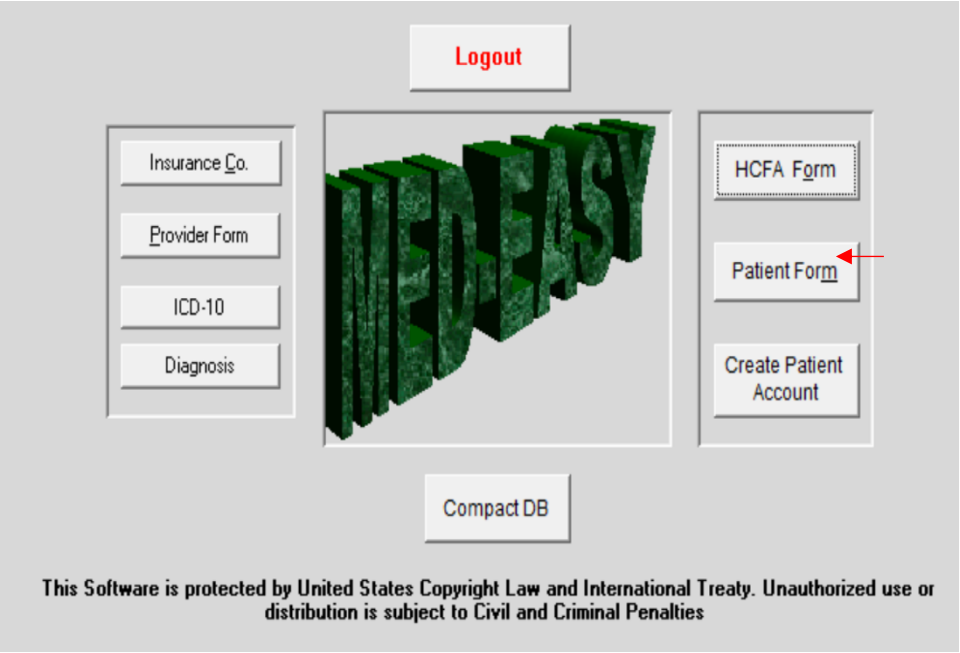
2.3.2 Main Menu

After logging into the system, you will be presented with the “Main Menu” of the system. This will have two options, one where you can access forms, such as “HCFA”, and another where you can input information. You will also have the option to “Log Out” of the system, which is located at the top of the “Main Menu” page. **(GUI02-3)**



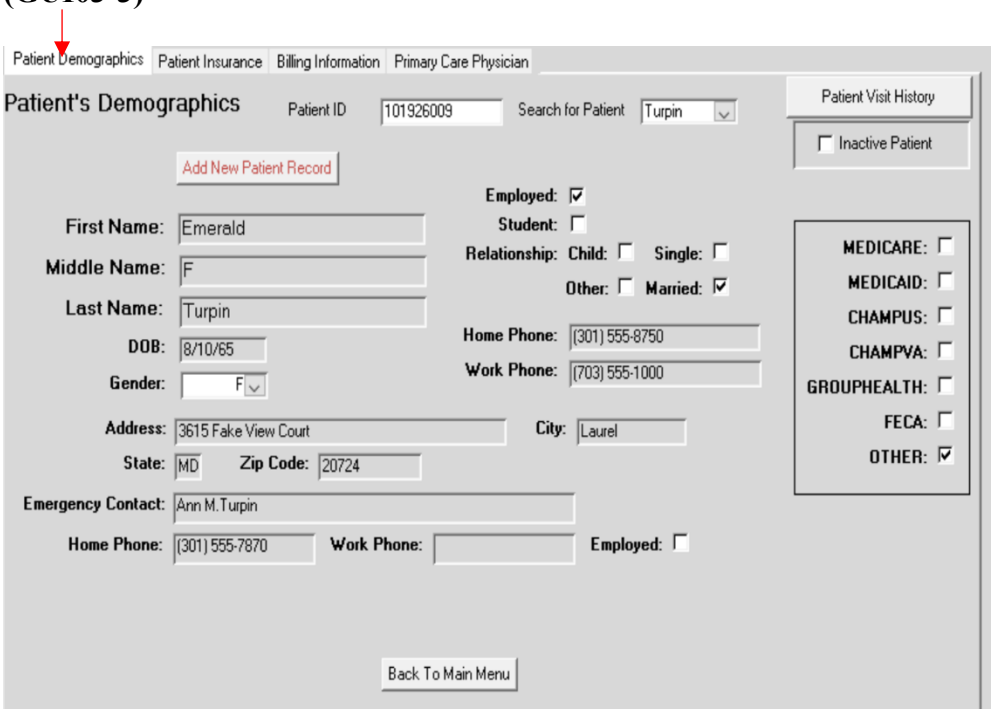
2.3.3 Patient Form

When selecting the “Patient Form” button, you will be directed to the “Patient Demographics” menu. In addition, you will be able to navigate to the patients “Insurance”, “Billing”, “Primary Physician”, and “Patient History” sections when desired. When you wish to move between pages, simply click the appropriate button located on the left hand side of the screen. When located on editable pages, you will click the information you would like to change, and you will have the ability to modify the fields as you see fit. The information is automatically saved when entered, so there is no extra work that is required to save the information. **(GUI02-3-1)**



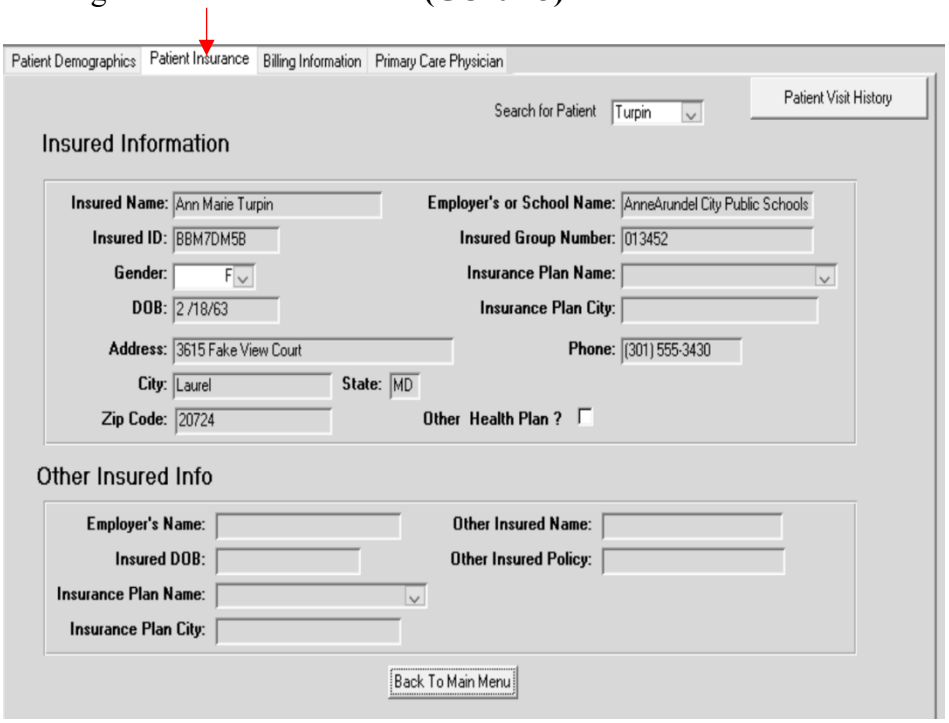
2.3.3.1 Patient Demographics

When located on this page, you will click the information you would like to change, and you will have the ability to correct or modify the given fields as you see fit. You may go back to the main menu at any time, simply by clicking “Back to Main Menu”. **(GUI03-3)**



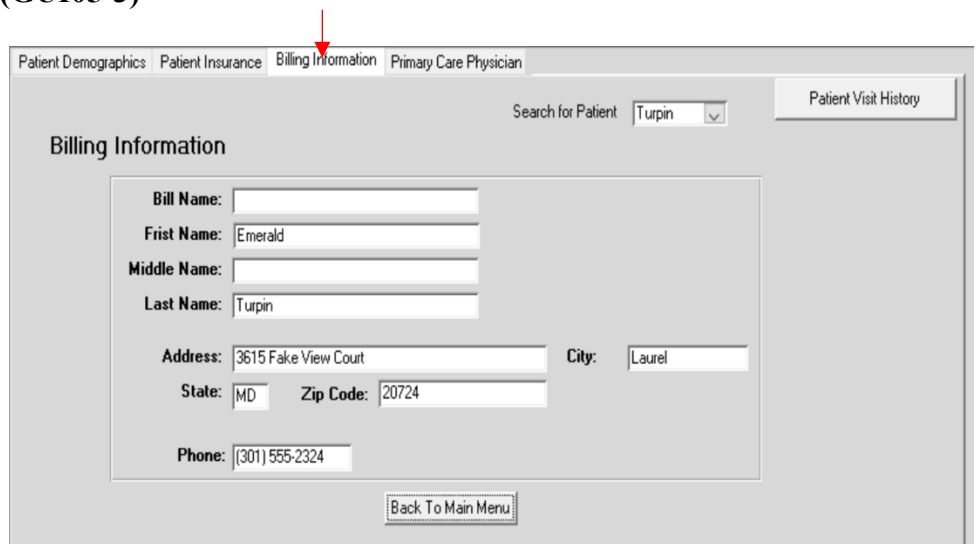
2.3.3.2 Insurance

Any changes can be made to the information on this page when needed. You may go back to the main menu at any time, simply by clicking “Back to Main Menu”. **(GUI04-3)**



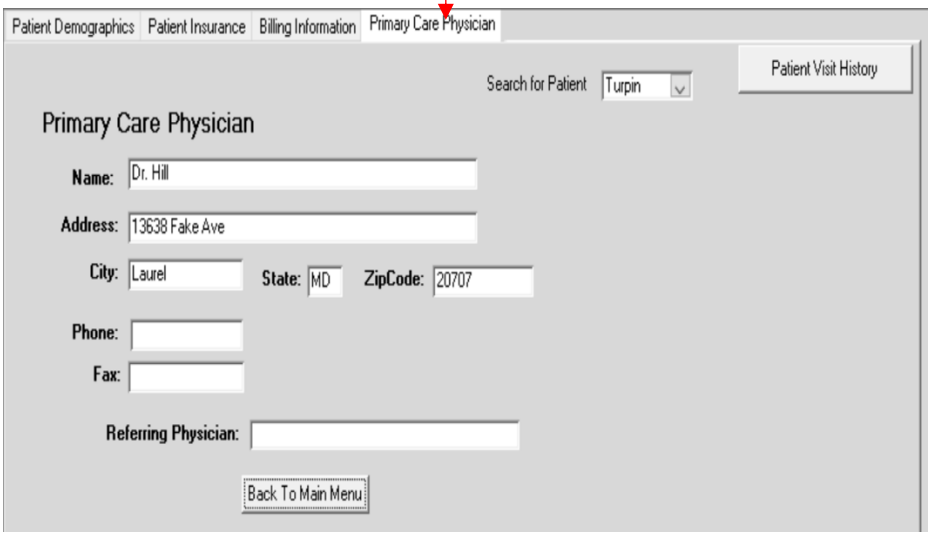
2.3.3.3 Billing

All fields are editable on this screen, and you may go back to the main menu at any time, simply by clicking “Back to Main Menu”. **(GUI05-3)**



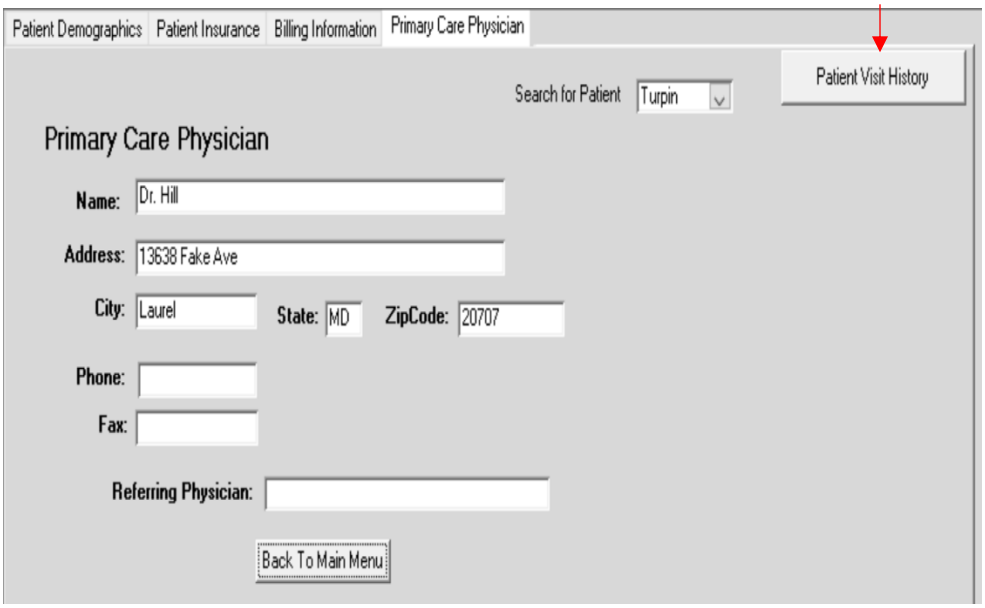
2.3.3.4 Primary Physician

You will have the ability to change all of the information on the page, and you may go back to the main menu at any time, simply by clicking “Back to Main Menu”. **(GUI06-3)**



2.3.3.5 Patient History

You may Log out of the system at any time, by clicking the “Log Out” button located at the bottom of the screen. When clicked, a display box will appear, so that a report may be generated. **(GUI07-3)**



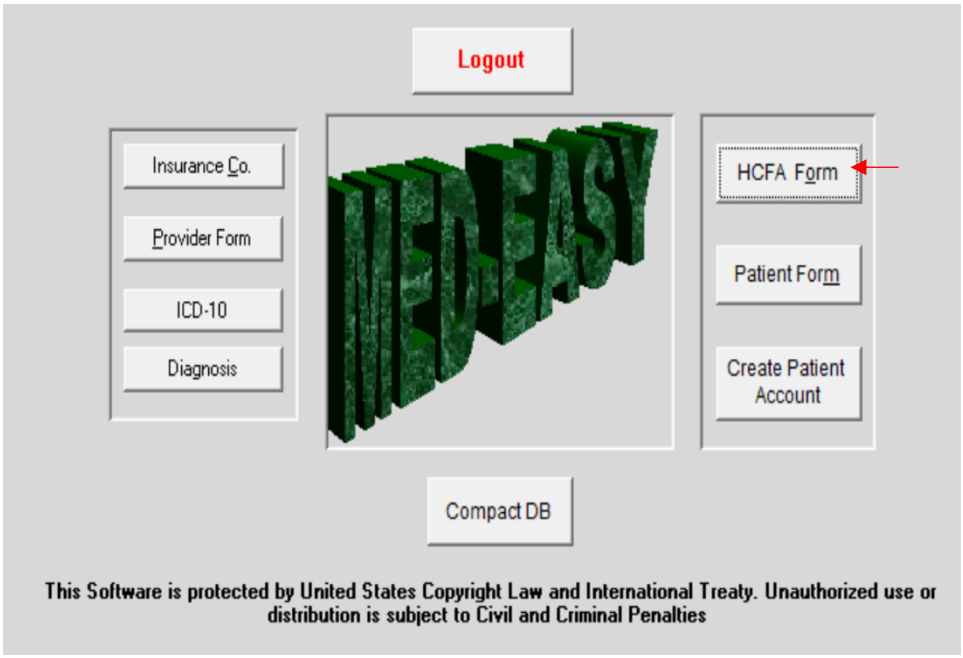
2.3.3.5.1 Patient Visit History

When the user is finished viewing their visit history, they will click the “x” in the upper right corner of the screen to go back to the last screen. **(GUI08-3)**

VisitID	PatientID	ProviderID	Date	Diagnosis-1	PinNO	Authorizatic	PlanName
1020790	212368774	EVELYN	4/11/16	311	528329-01		
1020791	Reddy, Sundeep S	EVELYN	6/28/16	296.90	528329-01	00QM20000	
1020793	Reddy, Sundeep S	EVELYN	7/4/16	296.90	528329-01	00QM20000	
* (New)			5/8/16				

2.3.4 HCFA

When selecting the “HCFA” button, you will be directed to the “HCFA” form, where you will have access to input all of the various information that is required within the form. **(GUI02-3-2)**



2.3.4.1 Navigation

Please click the desired button when entering the information. **(GUI09-3)**

2.3.4.1.1 Top Panel

2.3.4.1.1.1 Lookup Record

This option provides a search bar that will allow you to pull the HCFA forms for a specified patient. **(GUI010-3)**

2.3.4.1.1.2 Show All Records

This option provides all records for the selected individual from the “Lookup Visit/Patient”. **(GUI11-3)**

2.3.4.1.1.3 Add Record

This option generates a new HCFA Form when selected for an individual. **(GUI12-3)**

2.3.4.1.1.4 Delete Record

This option will remove the HCFA form from the system. **(GUI13-3)**

2.3.4.1.1.5 Add Patient

This option provides the ability to add a new patient so a HCFA Form can be created. **(GUI14-3)**

2.3.4.1.1.6 Menu

This option will take you back to the forms menu. **(GUI15-3)**

2.3.4.1.2 Entering Information

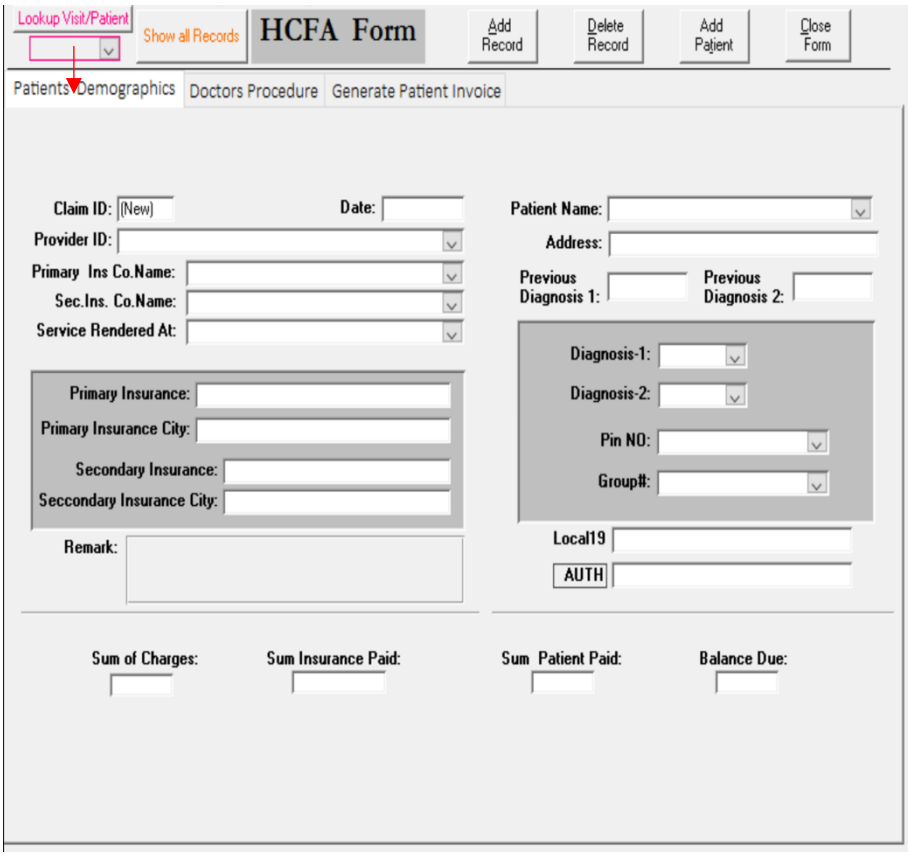
Once you have selected the correct patient, and have their records pulled, please enter the appropriate information into the various fields. **(GUI16-3)**

2.3.4.1.3 Entry Options

You may put information into three different areas, “Patient Demographics”, “Doctors Procedure”, or “Generate Patient Invoice”.

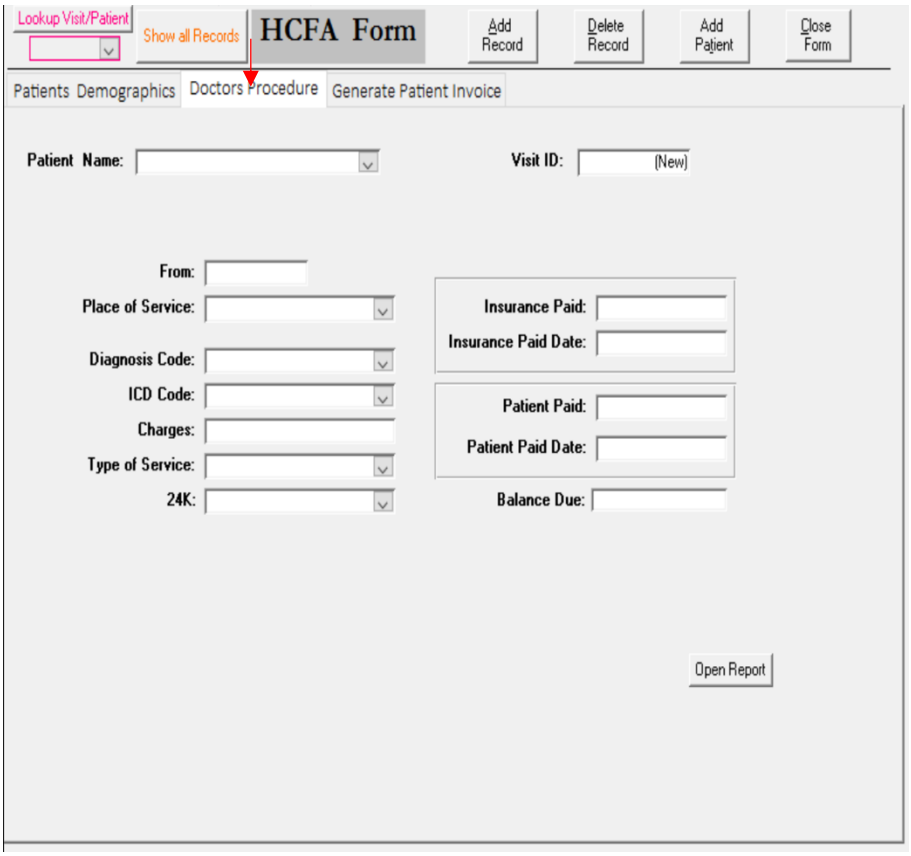
2.3.4.1.3.1 Patient Demographics

(GUI09-3-1)



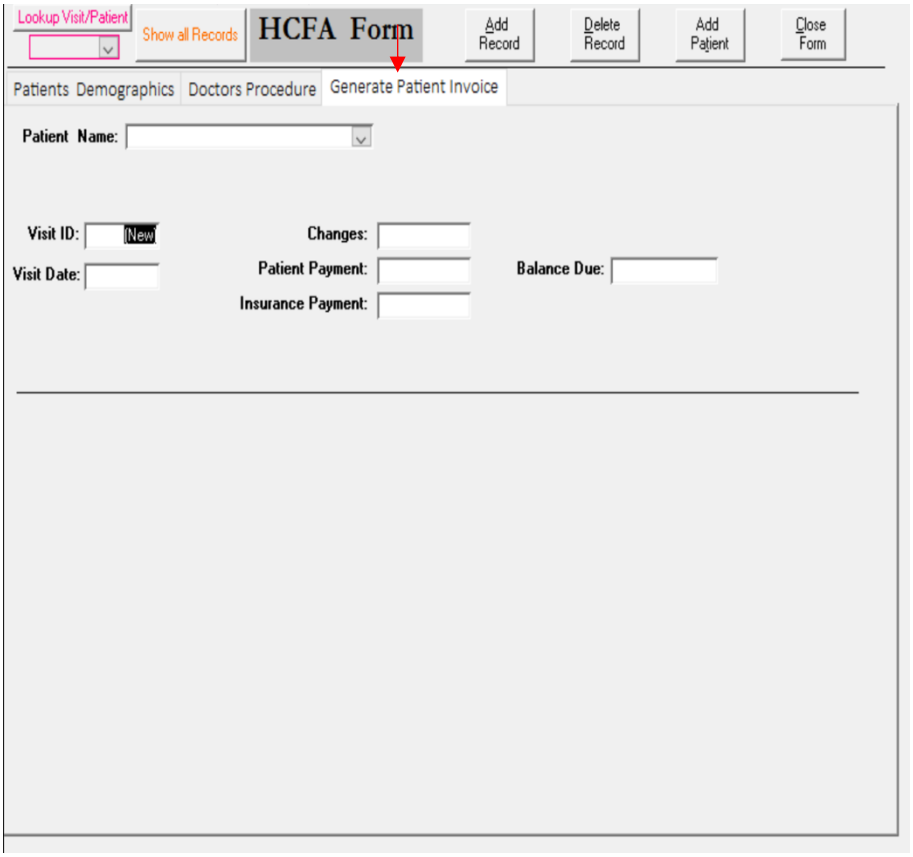
2.3.4.1.3.2 Doctors Procedure

This is where the doctor will enter the information from the previous visit. **(GUI17-3)**



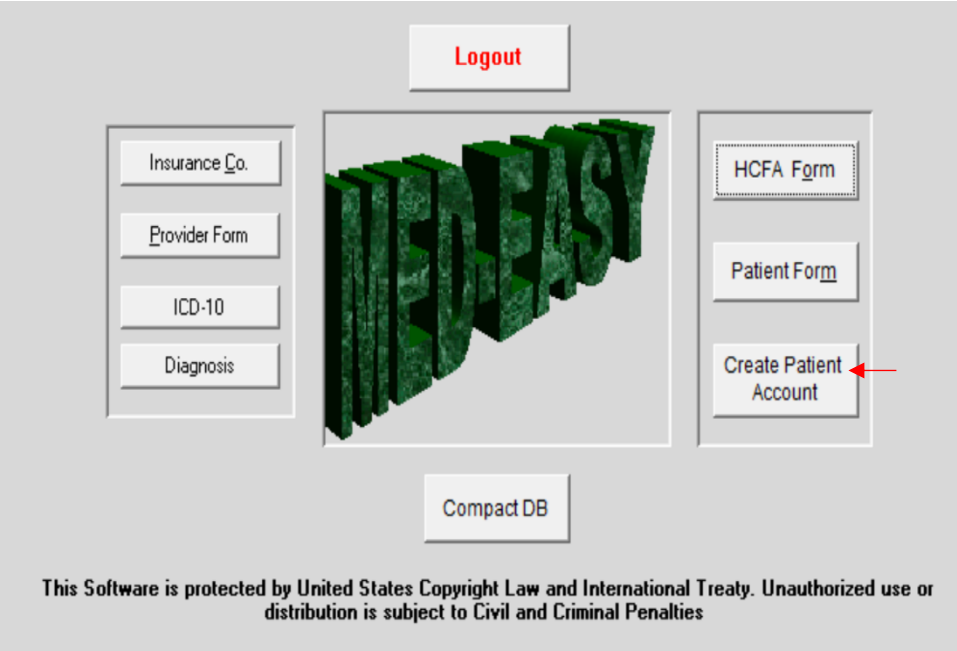
2.3.4.1.3.3 Generate Patient Invoice

This is the section of the program that creates the invoice for the various patients. **(GUI18-3)**



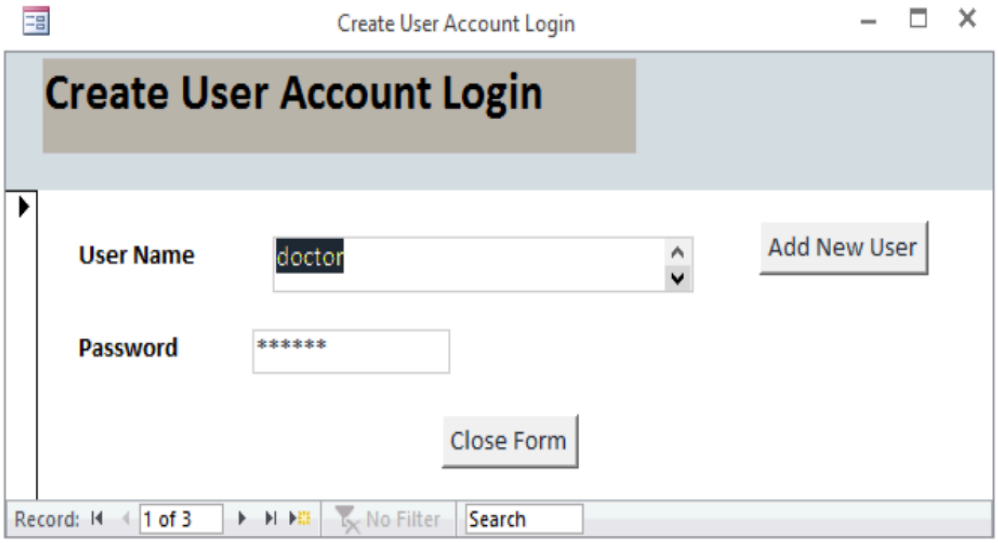
2.3.5 Create Patient Account

From the “Main Menu” screen, you will select “Create Patient Account”. **(GUI02-2-3)**



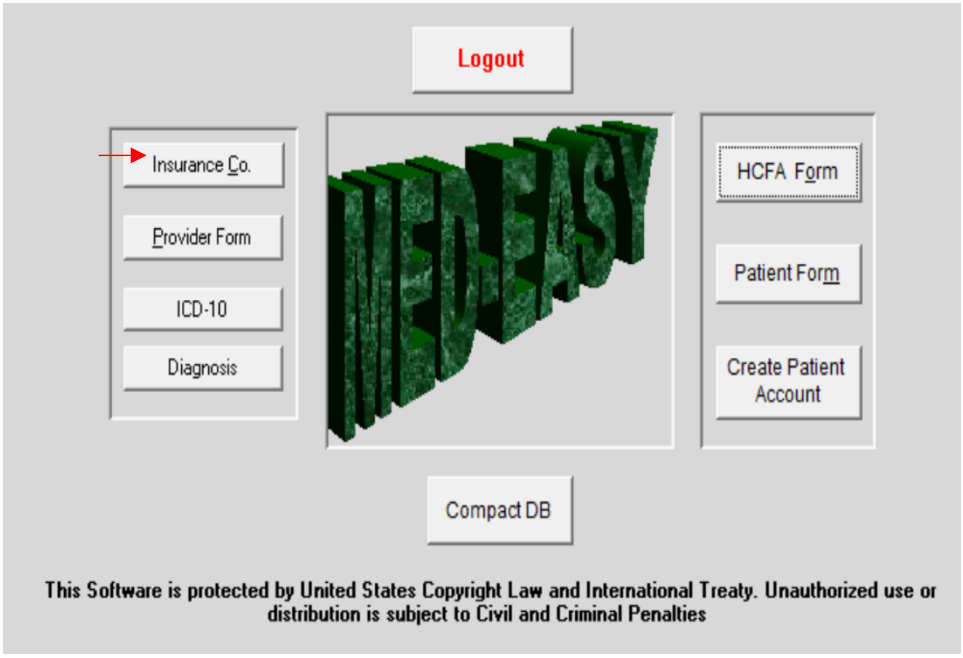
2.3.5.1 Data Entry

When on this page, the user will enter new users “username”, and “password”, and once completed, they will click “Add New User”. Once finished, the user may click “Close Form” to go back to the main menu. **(GUI19-2)**



2.3.6 Insurance Company

From the “Main Menu” screen, you will select “Insurance Co.”. **(GUI02-2-4)**

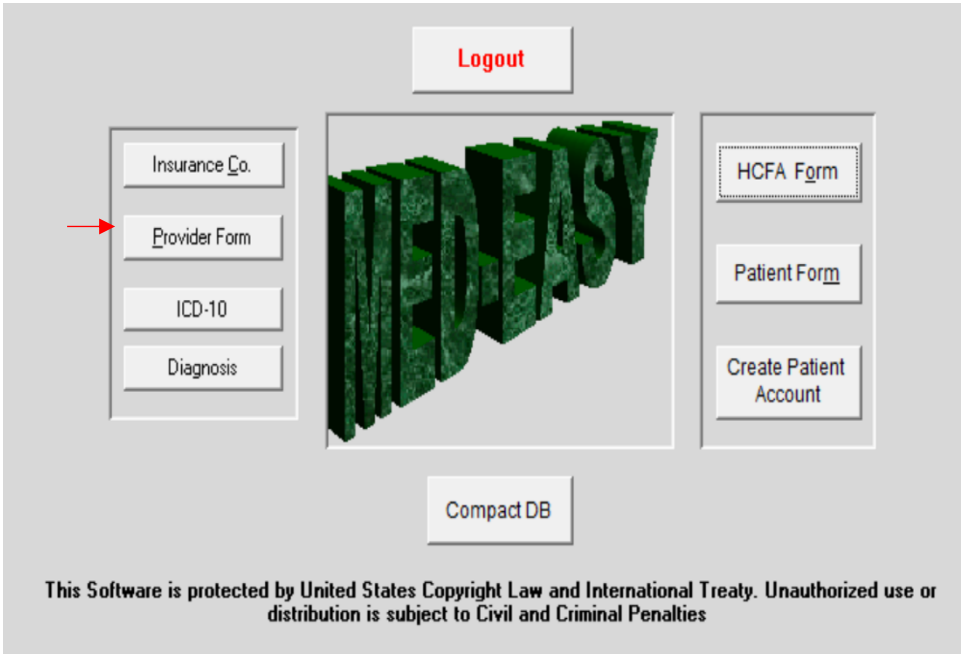


2.3.6.1 Data Entry

After selecting “Insurance Co.”, you will be brought to the screen below, where you will enter your information, and click “Add Record”. You will have the option to return to the main menu of the system, by clicking the button on the left hand side of the screen, named “Back to Main Menu”. **(GUI20-2)**

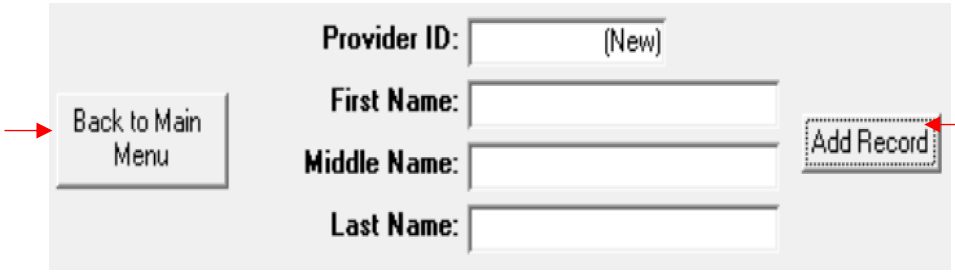
2.3.7 Provider Form

From the “Main Menu” screen, you will select “Provider Form”. **(GUI02-2-5)**



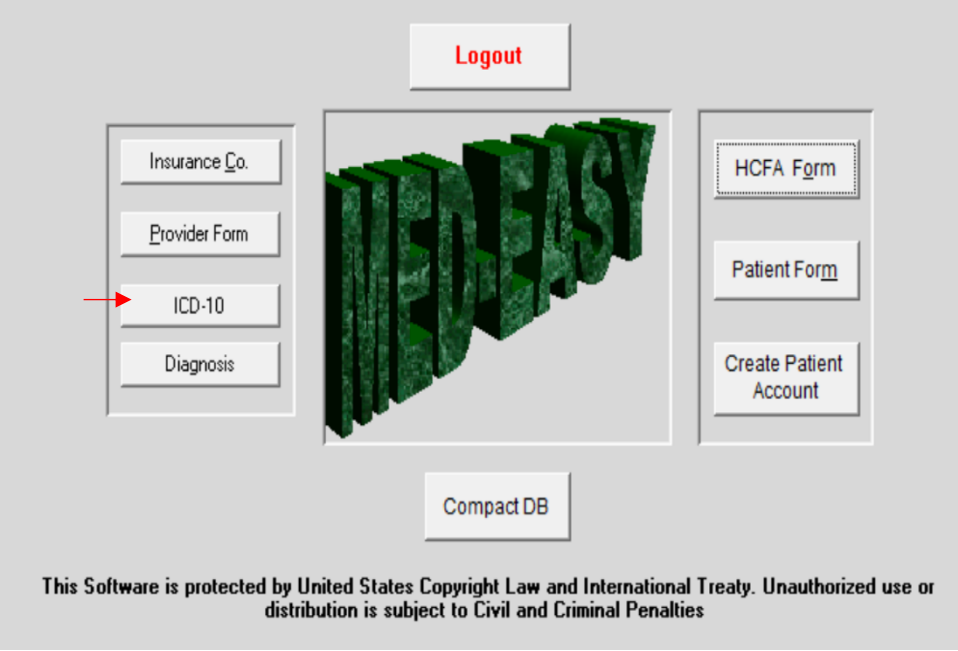
2.3.7.1 Data Entry

When you want to add a new provider into the system, you need to add the required information into the textboxes, and then click “Add Record” button, in order to save the new information into the system. You will them click “Back to Main Menu”, to go back to the main menu. **(GUI21-2)**



2.3.8 ICD-10

From the “Main Menu” screen, you will select “ICD-10”. **(GUI02-2-6)**

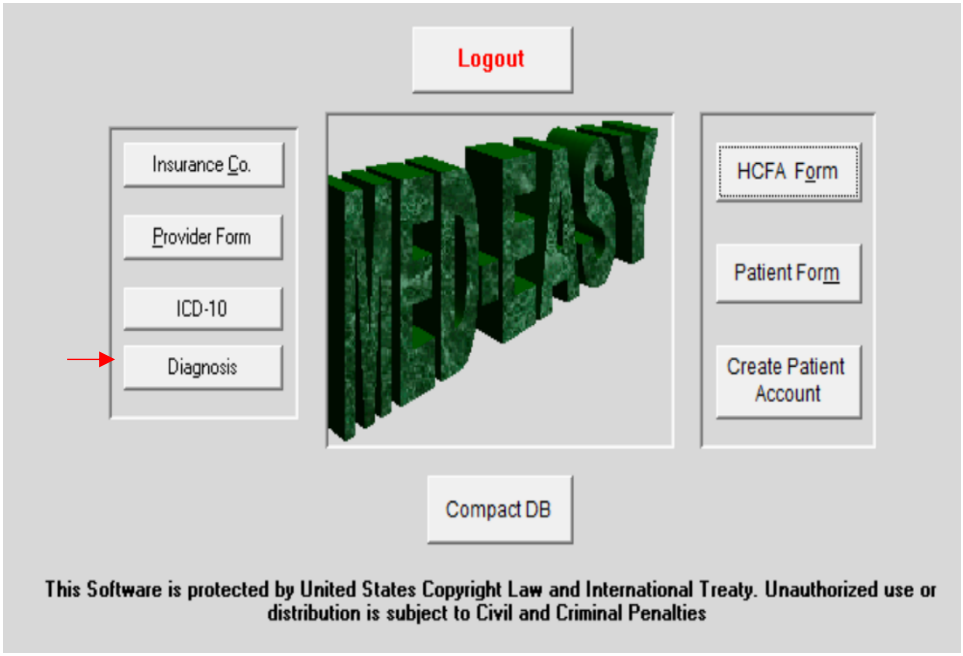


2.3.8.1 Adding Data

When adding a new ICD-10 code into the system, you will enter the information into the various textboxes, and click the “Add Record” button to save the new information into the system. You may then press the “Back to Main Menu”, to go back to the main menu. **(GUI22-2)**

2.3.9 Diagnosis

From the “Main Menu” screen, you will select “Diagnosis”. **(GUI02-2-7)**



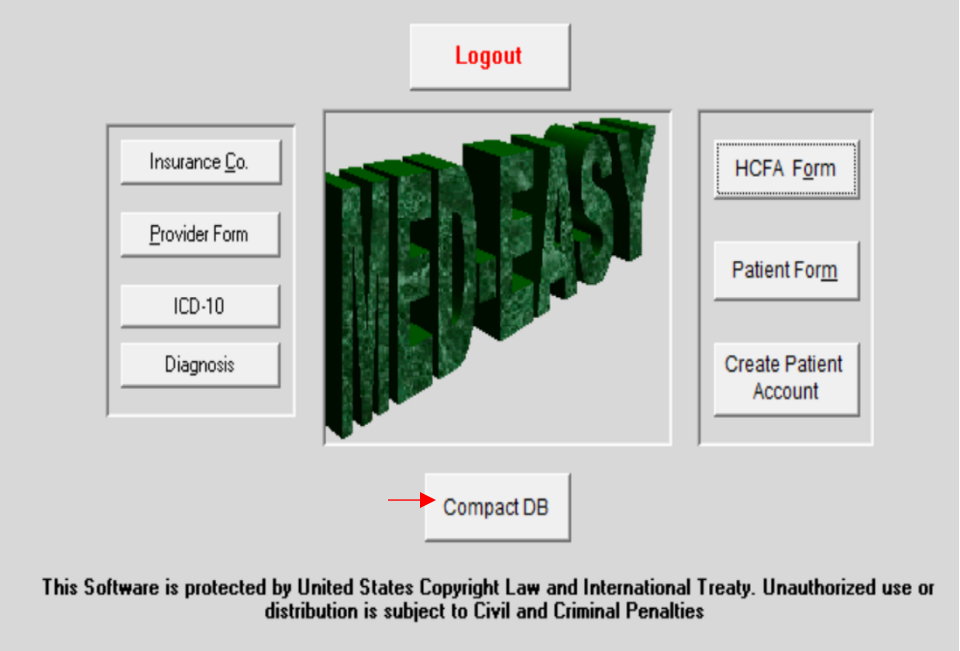
2.3.9.1 Data Entry

When you want to add new information into the system, you will enter the information into the various text boxes, and then click “Add Record” button to save the new information into the system. You may then press the “Back to Main Menu”, to go back to the main menu. **(GUI23-2)**

2.3.10 Compact Database

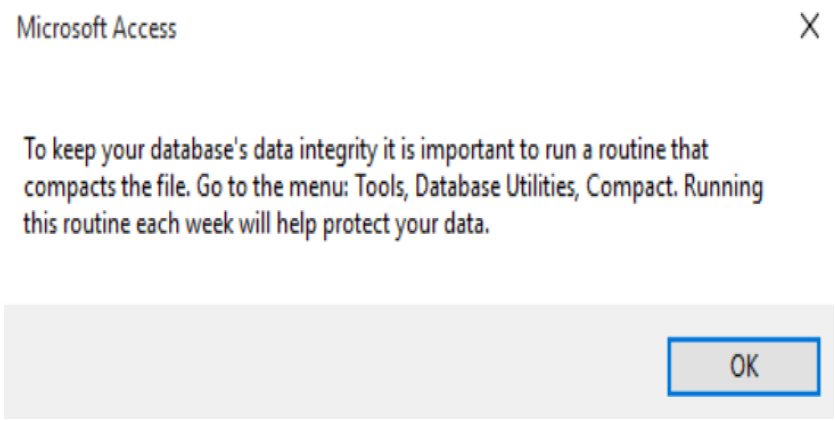
From the “Main Menu” screen, you will select “Compact Database”.

(GUI-2-2-8)



2.3.10.1 Display

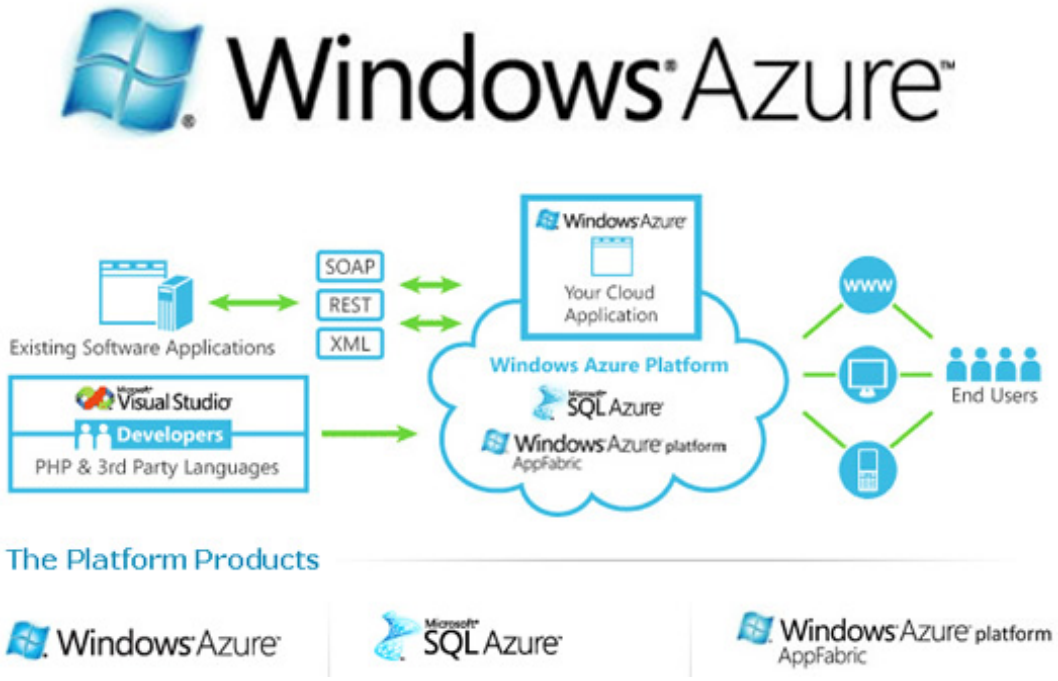
This button will only display a message to remind the primary users of the system to occasionally compact their database, and the necessary steps to accomplish it. **(GUI24-2)**



3. Deployment Diagram

In order for MedEasy to run properly, the client's system will need to have an Intel Core i5 Quad-core processor, with 25 GB of HDD, and an Intel HD Graphics card - or their respective equivalents. MedEasy will be a server side application, utilizing the Microsoft Azure cloud application. The existing software application (MedEasy) will be placed on the Azure Platform, and then be accessible in real time to the end users. The end users can be either a doctor, admin, or patient of the doctor's office.

Figure 10



This diagram can be found from the following link: http://techglimpse.com/wp-content/uploads/2014/03/azure1.jpg

4. Implementation of User Story

4.1 All Users

4.1.1 Log onto the System

UC1) As a user, I want to login my account, so that I can access the system.

4.1.2 Search for Patient

UC2) As an administrator, I want to check a patient account, so that I can know if the patient exists.

4.1.3 Produce a Claim

UC3) As an administrator, I want to get patient information, so that I can produce claim for them.

4.1.4 Update Account Balance

UC4) As an administrator, I want to update patients' account balance, so that they can pay for their remaining balance.

4.1.5 View Account

UC5) As a patient, I want to view my account, so that I can get my diagnosis report and invoice.

4.1.6 Make a Payment

UC6) As a patient, I want to make payments, so that I can paid off my balance on time.

5. Test Case

5.1 Test Case #1

Test Case ID		TC-01		
Test Case Description:		Verify the following text fields from the Login screen – Username, Password		
Related DFD		DFD 1	**Related sequence diagram**	SD-01

Precondition	**Post Condition**	**Test Data**
The Login GUI must exist	Patient login is successful	Test Data 1: **Username:** patient **Password:** patient
		Test Data 2: **Username**: pa1235 **Password:** 123124

Step #	**Step Details**	**Expected Results**	**Actual Results**	**Pass / Fail & comments**
1	Click Login	Error message: "Username is blank"		
2	Click 'OK' and 'Enter' *Test Data 1*	Login in to the system successfully		
3	Click 'Log Out'	Exit MedEasy		
4	Enter Test Data 2	Login failed		

5.2 Test Case #2

Test Case ID		TC-02		
Test Case Description:		Verify if user patient could view his/her account information		
Related DFD		DFD 2	**Related Sequence Diagram**	SD-03

Precondition	**Post Condition**	**Test Data**
The user must login	Patient demographics is accessible	Test Data 1: **Username:** patient **Password:** patient
The patient account interface existed	Patient insurance info & billing info is accessible	
	Patient history & payment is accessible	

Step #	**Step Details**	**Expected Results**	**Actual Results**	**Pass / Fail & comments**
1	Use Test Data 1 to login	Login successful		
2	Click the “Patient Demographics” tab	Display the ‘Patient Demographics’		
3	Click “Patient Insurance” tab	Display the ‘Patient's Insurance’ information		
4	Click the "Billing Information” tab	Display the ‘Patient's Billing’ information		
5	Click the “Primary Care Physician” tab	Display the patient's ‘Primary Care Physician’ information		
6	Click the “Patient History” tab 6) cursor position	Display the ‘Patient History’		
7	Click the "Payment" tab	Allow the patient to make a payment		
8	Click “Log Out”	Exit MedEasy		

5.3 Test Case #3

Test Case ID	TC-03		
Test Case Description:	Verify the text fields in the Payment screen		
Related DFD	DFD 7	**Related Sequence Diagram**	SD-07

Precondition	**Post Condition**	**Test Data**
The user must login, and the screen of "Payment" must exist	All fields in "Payment" have been filled	Test Data 1: **Username:** patient **Password:** patient
		Test Data 2: Name on Credit Card: Marry Lee EXP: 03/12/2018 Credit Card Number: 1234567890123456 CVC: 909 Amount Paid: $213.00

Step #	**Step Details**	**Expected Results**	**Actual Results**	**Pass / Fail & comments**
1	Use Test Data 1 to login	Login successful		
2	Click "Payment" tab	Display blank data fields		
3	Enter Test Data 2	Filled data fields		
4	Click Log Out	Exit MedEasy		

5.4 Test Case #4

Test Case ID		TC-04		
Test Case Description:		Verify the following text fields in the Login screen – Username, Password		
Related DFD		DFD 1	**Related sequence diagram**	SD-01

Precondition	**Post Condition**	**Test Data**
The Login GUI must existed	Admin or Doctor Login is successful	Test Data 1: **Username:** admin **Password:** admin
		Test Data 2: **Username:** we1235 **Password:** fsd3312
		Test Data 3: **Username:** doctor **Password:** doctor
		Test Data 4: **Username:** ascee3 **Password:** summer2016

Step #	**Step Details**	**Expected Results**	**Actual Results**	**Pass / Fail & comments**
1	Click “Login”	Error message: "Username is blank"		
2	Click “OK” and Enter Test Data 1	Login successful		
3	Click “Log Out”	Exit MedEasy		
4	Enter Test Data 2	Login failed		
5	Enter Test Data 3	Login successful		
6	Click “Log Out”	Exit MedEasy		
7	Enter Test Data 4	Login failed		
8				

5.5 Test Case #5

Test Case ID		TC-05		
Test Case Description:		Verify the text fields in the "Create Patient Account" and be able to save the filled information		
Related DFD		DFD 3	**Related Sequence Diagram**	SD-02

Precondition	**Post Condition**	**Test Data**
The user must login.	New patient account is created	Test Data 1: **Username:** admin/ **Password:** admin
The "Create Patient Account" button is able to guide the user to the GUI of "Create Patient Account"		

Step #	**Step Details**	**Expected Results**	**Actual Results**	**Pass / Fail & comments**
1	Use Test Data 1 to login	Login as an admin successfully		
2	Click "Create New Account"	Display "Create New Account"		
	Click "Add New User"	A blank text field page		
3	Input any username and password, and remember it, because you will need to use what you created to login again	Filled input is saved automatically		
4	Click “Back To Main Menu”	Main Menu		
5	Click “Log Out”	Exit MedEasy		
6	Login (use the username and password from step 3)	Error Message "Password Invalid" (because there are only three types of user accounts - patient, admin, and doctor. This means the username and password are to show the access level)		
7	Click “OK”	Login screen		

5.6 Test Case #6

Test Case ID	TC-06		
Test Case Description:	Update the text fields in the "Patient Form".		
Related DFD	DFD 3	**Related Sequence Diagram**	SD-05

Precondition	**Post Condition**	**Test Data**
The user must login, and the "Patient Form" is able to guide the user to next GUI	Data is updated	Test Data 1: **Username:** admin **Password:** admin
		Test Data 2: Address: 300 parker road Zip Code: 99999 Home Phone: 000-111-1234
		Test Data 3: Home Phone: 000-111-123456
		Test Data 4: Zip Code: 21210asd

Step #	**Step Details**	**Expected Results**	**Actual Results**	**Pass / Fail & comments**
1	Use Test Data 1 to login	Login as an administrator successfully		
2	Click the "Patient Form" button	Open "Patient Form" GUI		
3	Select a patient name randomly in the "Search for Patient" dropdown menu, under the "Patient Demographics" tab.	The selected patient information is displayed in all tabs, which includes the Patient Demographics, Patient Insurance, Billing Information, Primary Care Physician, and Patient History.		
4	Enter Test Data 2	Data is entered and saved		
5	Enter Test Data 3	User hears a sound, because the input data has to be exactly 10 digits long		
6	Click “OK” and Enter Test Data 4	Error message: "The value you entered isn't valid for this field."		
7	Click “OK” and Click "Back to Main Menu"	Error message: "The value you entered isn't valid for this field."		
	Delete "asd" in Zip Code text field and Click "Back To Main Menu"	Main Menu		
8	Click “Log Out”	Exit MedEasy		

5.7 Test Case #7

Test Case ID	TC-07		
Test case description:	Add record in the "Patient Form" screen		
Related DFD	DFD 3	**Related sequence diagram**	SD-05

Precondition	**Post-condition**	**Test Data**
The user must login, and the "Patient Form" is able to guide the user to to the next GUI	New patient record is created	Test Data 1: **Username:** admin **Password:** admin
		Test Data 2: First Name: Marry Last Name: Lee DOB: 1/1/1988 Gender: F Emergency Contact: 222-333-4444
		Test Data 3: Address: 300 parker road Zip Code: 1234567

Step #	Step Details	Expected Results	Actual Results	Pass / Fail & comments
1	Enter Test Data 1 to login	Login as an administrator successfully		
2	Click the "Patient Form" button	Open the "Patient Form"		
3	Click the "Add New Patient Record" button	Blank page appears		
4	Enter Test data 2 to fill the mandatory fields	Input is saved		
5	Enter Test data 3	Error message: "The value you entered isn't valid for this field."		
6	Click “OK” and Click "Back to Main Menu"	Error message: "The value you entered isn't valid for this field."		
7	Delete "67" in the Zip Code text field and Click the "Back To Main Menu" button	Main Menu		
8	Click Log Out	Exit MedEasy		

5.8 Test Case #8

Test Case ID	TC-08		
Test Case Description:	Validate if a doctor or admin could view patient information		
Related DFD	DFD 2	**Related sequence diagram**	SD-03

Precondition	**Post-condition**	**Test Data**
The user is authorized and the patient exists	The patients information is viewed by a doctor or an administrator	Test Data 1: **Username:** doctor **Password:** doctor
		Test Data 2: **Username:** admin **Password:** admin

Step #	**Step Details**	**Expected Results**	**Actual Results**	**Pass / Fail & comments**
1	Enter Test Data 1 to login	Login as a doctor successfully		
2	Click the "Patient Form" button	The "Patient form" appears		
3	Find the patients name in "Search for Patient" dropdown menu in "Patient Demographics" tab.	The patient information is displayed on all tabs, which includes the Patient Demographics, Patient Insurance, Billing Information, Primary Care Physician, and Patient History.		
4	Click the “Back To Main Menu” button	Main Menu		
5	Click “Log Out”	Exit MedEasy		
6	Enter Test Data 2 to login	Login as an administrator successfully		
7	Click the "Patient Form" button	The "Patient form" appears		
8	Find the patients name in "Search for patient" dropdown menu, in "Patient Demographics" tab.	The patient information is displayed on all tabs, which includes the Patient Demographics, Patient Insurance, Billing Information, Primary Care Physician, and Patient History.		
9	Click the “Back To Main Menu” button	Main Menu		
10	Click “Log Out”	Exit MedEasy		

5.9 Test Case #9

Test Case ID	TC-09		
Test Case Description:	Verify the text fields in the "Generate Patient Invoice" and the derived value field will be calculated automatically		
Related DFD	DFD 6	**Related Sequence Diagram**	SD-06

Precondition	**Post Condition**	**Test Data**
The user is authorized, and GUI of the HCFA Form exists	Invoice is produced and stored	Test Data 1: **Username:** admin **Password:** admin
		Test Data 2: Charges: 213 Patient payment: 10 Insurance payment: 203
		Test Data 3: Charges: asdf

Step #	**Step Details**	**Expected Results**	**Actual Results**	**Pass / Fail & comments**
1	Enter Test Data 1 to login	Login as an administrator successfully		
2	Click the "HCFA Form" button	The "HCFA form" page appears		
3	Click the "Generate Patient Invoice" tab	The "Generate patient invoice" tab appears		
4	Click the "Patient Name" to find a patient randomly in the dropdown menu	Found a patient.		
5	Enter Test Data 2	"Balance Due" is calculated automatically		
6	Enter Test Data 3	Error message: "The value you entered isn't valid for this field."		
7	Click “OK” and Click "Close Form"	Error message: "The value you entered isn't valid for this field."		
	Delete Test Data 3 and Click "Close Form"	Main Menu		
8	Click Log Out	Exit MedEasy		

5.10 Test Case #10

Test Case ID	TC-10		
Test Case Description:	Generate report		
Related DFD	DFD 5	**Related Sequence Diagram**	SD-04

Precondition	**Post-condition**	**Test Data**
The user has been authorized, and the HCFA Form exists and searched patient is exists	HCFA form is ready	Test Data 1: **Username:** admin **Password:** admin

Step #	**Step Details**	**Expected Results**	**Actual Results**	**Pass / Fail & comments**
1	Use Test Data 1 to login	Login as an administrator successfully		
2	Click the "HCFA Form" button	The "HCFA form" page appears		
3	Click the "Patient Demographics" tab	The "Patient Demographics" tab appears		
4	Click the "Patient Name" to find the patients name in the dropdown menu	The patient has been found. All three of the tabs appear, with the results from the same patient.		
7	Click "Close Form"	Main Menu		
8	Click “Log Out”	Exit MedEasy		

5.11 Test Case #11

Test Case ID		TC-11		
Test Case Description:		Validate if a doctor be able to produce patient diagnosis and ICD-10		
Related DFD		DFD-4	**Related Sequence Diagram**	SD-08

Precondition	**Post-condition**	**Test Data**
The user has been authorized, and the HCFA Form button is able to guide the user to the HCFA Form, with all of the text fields in "Doctors Procedure" tab being populated	The diagnosis and ICD-10 codes have been filled and saved	Test Data 1: **Username:** doctor **Password:** doctor
		Test Data 2: Place of Service: Maryland General Hospital Diagnosis code: 1 ICD code: 88803

Step #	Step Details	Expected Results	Actual Results	Pass / Fail & comments
1	Enter Test Data 1 to login	The doctors login is successful		
2	Click the "HCFA form" button	The "HCFA form" page appears		
3	Click the "Doctors Procedure" tab	The "Doctors Procedure" tab appears, and allows the doctor to input/select "Place of Service", "Diagnosis Code" and "ICD-10 Code", etc		
4	Click the "Patient Name" to find the patient in the dropdown menu	The patient is found		
5	Enter Test Data 2	Data in entered and saved		
6	Click “Close Form”	Main Menu		
7	Click “Log Out”	Exit MedEasy		

6. Traceability Analysis

Use Case	GUI	DFD	User Manual	Test Case
UC01	GUI01-1, GUI01-2, GUI01-3	DFD1	2.1.1, 2.2.1, 2.3.1	TC-01 TC-04
UC02	GUI03-3, GUI09-3, GUI17-3, GUI18-3	DFD1, DFD2, DFD3	2.3.3.1, 2.3.4.1, 2.3.4.1.3.2, 2.3.4.1.3.3	TC-05
UC03	GUI02-3-2, GUI09-3, GUI09-3-1, GUI17-3, GUI18-3	DFD1, DFD2, DFD5	2.3.4, 2.3.4.1, 2.3.4.1.3.1, 2.3.4.1.3.2, 2.3.4.1.3.3	TC-09
UC04	GUI18-3	DFD1, DFD2, DFD6	2.3.4.1.3.3	TC-08
UC05	GUI06-1, GUI07-1, GUI08-1, GUI09-1	DFD1, DFD2, DFD6	2.1.6, 2.1.6.1, 2.1.7, 2.1.7.1	TC-06
UC06	GUI06-1, GUI07-1	DFD1, DFD2, DFD7	2.1.6, 2.1.6.1	TC-03