ABSTRACT

This project report describes the application of Management System for Patient Information and Insurance Billing (MSPIIB) in a physician’s office. This software is prepared for Dr. Geeta Lele, M.D., practicing medicine at Hobbs, NM. Dr. Lele expressed a desire for a flexible and easy to access patient information database and Electronic Insurance claim processing software system. This software module is built using a user interface and a background database. MSPIIB is easy to access from anywhere, as it is an Internet based system. The use of MSPIIB results in ready access to patients’ information and electronic claims processing.
# TABLE OF CONTENTS

Abstract ........................................................................................................................................... ii

Table of Contents ........................................................................................................................... iii

List of Figures ................................................................................................................................. vi

1. Background and Rationale ........................................................................................................ 1

   1.1 Medical Insurance Billing .......................................................................................... 2
       1.1.1 Paper Insurance Claim Process .................................................................... 2
       1.1.2 Electronic Claims Processing ......................................................................... 3

   1.2 Patient Information Database .................................................................................. 4

2. Narrative ..................................................................................................................................... 5

   2.1 Overview .................................................................................................................. 5

   2.2 Step-by-Step Process of MSPIIB ........................................................................... 5
       2.2.1 Provider Selection .......................................................................................... 6
       2.2.2 Provider Information .................................................................................... 7
       2.2.3 Procedure Code ............................................................................................. 8
       2.2.4 Diagnosis Code .............................................................................................. 9
       2.2.5 Insurance Carrier Records .......................................................................... 10
       2.2.6 Patient Information ....................................................................................... 11
       2.2.7 Electronic Receiver Records ......................................................................... 12
       2.2.8 Security Setup ............................................................................................... 13

3. System Design ......................................................................................................................... 17

   3.1 Main Components of the Project .............................................................................. 17

   3.2 System Requirements ................................................................................................. 17
3.3 High Level Programming Languages .............................................18

3.3.1 PHP .....................................................................................18

3.3.2 MySQL ...............................................................................18

3.3.3 Perl ......................................................................................19

3.3.4 Java Script ...........................................................................19

3.4 Database Design .............................................................................19

3.4.1 Design and Entity Relationship .........................................19

3.4.2 Description of Database Tables ..........................................20

3.4.3 Entity Relationship of Log-in Process .......................23

3.5 User Interface Design .................................................................24

3.5.1 Transaction Module ............................................................24

3.5.2 Patients Module ..................................................................26

3.5.3 Diagnosis Module ...............................................................27

3.5.4 Procedures Module .............................................................27

3.5.5 Electronic Media Claims (EMC) .......................................28

3.5.6 Insurance Module.................................................................28

3.5.7 Provider Module .................................................................29

3.5.8 Safety Module.....................................................................29

3.5.9 Backup Module...................................................................30

3.5.10 Exit Module ........................................................................30

4. Evaluation and Results................................................................................31

5. Future Work ..........................................................................................34

6. Conclusion ..........................................................................................35
LIST OF FIGURES

Figure 2.1.  Step-by-Step Process of MSPIIB  6
Figure 2.2.  Provider Selection Page  7
Figure 2.3.  New Provider Page  8
Figure 2.4.  New Procedure Code Page  9
Figure 2.5.  New Diagnosis Code Page  10
Figure 2.6.  New Insurance Carrier Page  11
Figure 2.7.  New Patient Information Page  12
Figure 2.8.  New EMC Receiver Page  13
Figure 2.9.  User Login Page  14
Figure 2.10.  Add User Page  15
Figure 2.11.  Change Password Page  16
Figure 3.1.  Entity Relationship Diagram of MSPIIB  20
Figure 3.2.  Schematic Diagram for Login Process  23
Figure 3.3.  Schematic Diagram for Transaction Process  26
1. BACKGROUND AND RATIONALE

Federal Law “Health Insurance Portability And Accountability Act” (HIPAA) [HHS 2003] mandates physicians to maintain a file for every patient visiting their office. This file consists of general information, the nature of the visit, medical insurance information, and method of payment for the services rendered. Patient information is collected on paper and stored in filing cabinets. Patient files are numerous, so special filing procedures and techniques were created for easy retrieval and future referencing.

The insurance claims process involves the use of paper forms to record general information, patient medical diagnosis information, and their subsequent submission to insurance companies. A physician often uses CPT (Current Procedural Terminology) and ICD (International Classification of Diseases) codes to communicate patient medical diagnosis [AMA-ASSN 2003]. The support staff uses the CPT and ICD codes to prepare an invoice with charges and the information is sent to the insurance company by mail. Each claim normally takes about 90 days for approval and all unapproved claims are filed separately and resubmitted.

The existing claims process is not streamlined and as a result, many claims go unaccounted for and unpaid. Difficulties involved in maintaining patients records and insurance claims have provided an opportunity to develop a software package to streamline the process. This project is a software package with a database for CPT, ICD, insurance records, and patient information. Physicians and their support staff can use the software package to store patient information and send insurance claims electronically. This software package may also benefit insurance companies.
1.1 Medical Insurance Billing

When a physician treats a patient, the doctor’s office must file an insurance claim to get paid for their services. This claim is usually filed on paper and sent by mail. These paper claims are notoriously slow, often taking 60-90 days or more for the doctor to get paid.

While filing a claim, the support staff uses the procedural and diagnosis codes. Procedural and diagnosis codes are the communication language of the healthcare profession. Codes used by providers, insurance carriers, and third party payers clearly communicate specific health conditions. Codes also standardize and facilitate the description of healthcare procedures and diagnoses. Procedure codes, called CPT (Current Procedural Terminology) codes, were established by the American Medical Association. Diagnosis Codes are called ICD (International Classification of Diseases) codes, and are published by the World Health Organization [AMA-ASSN 2003].

Procedure codes are used to create claims for billing patients and insurance carriers. The general process for a paper claim is described in section 1.1.1.

1.1.1 Paper Insurance Claim Process

Following are the steps involved in a typical paper insurance claim process:

**Step 1**: The physician or an office manager creates an invoice with charges often called a “superbill” by the healthcare industry.

**Step 2**: Information from the invoice is manually transferred to the paper claim form HCFA 1500 (The Health Care Financing Administration) [DOL 2003]. This process is very time consuming.

**Step 3**: The claims are sent to respective patient insurance companies.
Step 4: The claims are entered into the insurance company’s system for review, along with any supporting documentation.

Step 5: Finally, the claim is accepted for payment, delayed, or rejected. Currently, about 30% of all paper claims are turned down because of improper coding or lack of proper information. When a claim is rejected, it’s usually up to the office manager to follow through with the insurance company. He or she must take the necessary steps to correct the claim and resubmit it to the insurance company. This is very time consuming for the physician’s staff. The claim is then subject to the entire process once more.

Step 6: Once all the above steps are completed and the claim is approved, the EOB (explanation of benefits) and payment are sent back to the physician’s office. The processing of paper claims may take 60 days, 90 days or longer, delaying reimbursement to the physician.

1.1.2 Electronic Claims Processing

Electronic claims processing allows these claims be processed efficiently, saving the healthcare provider time and money. With electronic claims processing, payment time is drastically reduced to 7 to 21 days on average. Various steps involved in electronic filing by a third party are as follows:

Step 1: The physician gives the billing company all patient information and superbills (an internal form that lists services rendered and charges made like an invoice).

Step 2: The billing company electronically transmits the physician’s data to the clearinghouse.

Step 3: The clearinghouse electronically forwards the edited claims to the insurance payer.
Step 4: The insurance payer sends the payment (reimbursement) directly to the physician. MSPIIB provides custom software to perform electronic claim processing and save time and expenses.

1.2 Patient Information Database:

In a realistic world, physicians diagnose a number of patients in a day. In order to track patients, their personal records, and diagnoses are stored in a secure location. Patient personal information consists of name, gender, date of birth, home address, etc. Diagnoses list ailments like high blood pressure, diabetes, etc.

Federal Law “Health Insurance Portability and Accountability Act” [HHS 2003] (HIPAA) mandates physicians to keep patients records in a secure location so that the records do not become public. To comply with the law, physicians file patient’s records in a paper format and keep them in a secure location inaccessible to the public. Over the years the filing stack becomes large and any attempt to access individual patient information becomes a tedious task.

Conventional paper filing of patient’s records can be replaced by the use of a database built to store all the information electronically. Any unauthorized access can be prevented by secure login pages thus complying with the legal regulations.

An electronic database will reduce manpower and save time in accessing individual records.
2. NARRATIVE

2.1 Overview

Management System for Patient Information and Insurance Billing (MSPIIB) is designed to make a physician's practice run more efficiently in terms of storage and retrieval of patient data, and referral tracking. It enables the healthcare provider to quickly create multiple reports for practice management. It helps the physician develop sound financial practices. Electronic storage of practice, patient, and transaction data files will enable more efficient filing of claims and faster payment to the practice.

2.2 Step-by-Step Process of MSPIIB

Health Care Financing Administration forms are used by non-institutional providers and suppliers to bill Medicare. They are also used for billing Medicaid and private insurance covered services.

The flow chart in Figure 2.1 [Buck 1998] shows an outline of the sequence for generating an Insurance claim. Each step is shown within an oval, followed by the data fields needed to complete that window. The process results in gathering and entering of data for the HCFA-1500 claim form [DOL 2003].
2.2.1 Provider Selection

The Provider Selection Page as shown in Figure 2.2 below enables the user to select a provider in a practice. This software can be used in a clinic with several providers, or with single doctor.
2.2.2 Provider Information

The New Provider Information Page, as shown in Figure 2.3 below, enables the entry of the provider's data such as Name, Credentials, Address, City etc. For a list of all data fields see table providers in Appendix A. There are search, edit and delete functions for previously entered providers.
2.2.3 Procedure Codes

Procedure codes are used to communicate procedure information between patient, provider and third-party payers. Figure 2.4 is a snapshot of the New Procedural Code Page. In the New Procedural Code Page, users can enter a new code, description, code type, type of service, place of service, default modifiers and charged amounts.

Existing codes can be searched using the search option. There are edit and delete options for existing codes. An upload function can be used to enter a large number of existing codes into the system.
2.2.4 Diagnosis Codes

Each patient is treated with a procedure. The reason for providing a particular service or treatment to a patient is identified by diagnosis code. In effect, the procedure code tells what the doctor did and the diagnosis code tells what the doctor found. Figure 2.5 is a snapshot of the New Diagnosis Code Page. In the New Diagnosis Code Page, users can enter a new code and description. An upload function can be used to enter a large number of existing codes into the system. There are search, edit and delete functions for existing diagnosis codes.
2.2.5 Insurance Carrier Records

Figure 2.6 is a snapshot of the New Insurance Carrier Page. The New Insurance Carrier Page allows the user to enter information regarding new insurance carriers like Name, Street, City and State etc. For a list of all data fields used see table insurance in Appendix A. There are search, edit and delete functions available for already existing insurance carriers.
2.2.6 Patient Information

Figure 2.7 is a snapshot of the New Patient Information Page. New patient details such as patient’s last name, first name, middle name etc., are entered in this screen. For a list of all the data fields available, see patient table in Appendix A. This screen also has search, edit and delete functions for existing patients.
2.2.7 Electronic Receiver Records

This window provides options to send claims electronically and to set up a new Electronic Media Claims (EMC) receiver record. Figure 2.8 is a snapshot of the New EMC Receiver Page. To set up a new EMC receiver record, information like receiver name and e-mail address has to be entered. This page also provides an option to search, delete and edit existing EMC receiver records. For a list of all the data fields available see emc table in Appendix A.
2.2.8 Security Setup

Security Setup feature protects the integrity and privacy of information files and also safeguards the data from unauthorized access. Each user is assigned a username and password for system access. User authorization is required to use the system. To ensure safety, a **User Login Page** as shown in Figure 2.9 is provided to access the system which prompts for username and password. The system validates the user's authentication and admits the user into the system. Otherwise, it displays user invalid or password invalid message.
Figure 2.9: User Login Page.

Only authorized personnel such as a system administrator can add new users to the system. Figure 2.10 is a snapshot of the Add User Page.
Figure 2.10: Add User Page.

As shown in Figure 2.11 existing users can change their password in Change Password Page.
Figure 2.11: Change Password Page.
3. SYSTEM DESIGN

Management System for Patient Information and Insurance Billing (MSPIIB) is a Linux based system, developed using PHP (Hypertext Preprocessor), MySQL, and Perl.

3.1 Main Components of the Project

The following are the major components of MSPIIB system:

- User authentication page,
- Provider, patient, and insurance information modules,
- Interface to add, update, search, and maintain database,
- PHP scripts for database queries,
- Perl module for report generation,
- System backup, and
- Mailing script for electronic transmittal of patient insurance claims.

3.2 System Requirements

The following are the requirements for successful deployment of MSPIIB:

- Compatible PC with at least 128MB RAM and a hard drive,
- PC with PHP and Perl programming languages installed,
- Web Authoring tools like Internet Explorer, Netscape Navigator, and
- MySQL database.
3.3 High Level Programming Languages

High level programming languages such as PHP (Hypertext Preprocessor), MySQL, HTML (Hyper Text Markup Language) and Perl were utilized in the development of the MSPIIB application.

3.3.1 PHP (Hypertext Preprocessor)

MSPIIB uses PHP, a server-side scripting language designed specifically for the Web applications. PHP code is embedded within the HTML page and it is executed each time the page is visited. The PHP code is interpreted by the Web server and generates the HTML or other output that the user sees. PHP has many advantages. Some of them are [Yank 2001]:

- High performance,
- Interface to many different database systems,
- Built-in libraries for many common web tasks,
- Low cost,
- Ease of learning and use,
- Portability and
- Availability.

Because of the above features PHP has been chosen for the server side scripting [Weiling. L 2001].

3.3.2 MySQL

MySQL was chosen because it is a very fast, robust, relational database management system (RDBMS). The MySQL server controls access to the data to ensure
that multiple users can work with it concurrently, to provide fast access to it, and ensure
that only authorized users can obtain access. MySQL is a multi-user, multi-threaded
server. It uses SQL (Structured Query Language), the standard database query language
worldwide [MYSQL 2003a].

3.3.3 Perl (Practical Extraction and Reporting Language)

Perl was chosen because it is an interpreted language optimized for scanning
arbitrary text files, extracting information from those text files, and printing reports based
on that information. It's also a good language for many system management tasks. The
language is intended to be practical (easy to use, efficient, complete) rather than beautiful
(tiny, elegant, minimal) [PERL 2003].

3.3.4 Java Script

The system uses the JavaScript executed on the client’s Web browser for the
validation of the user input data.

3.4 Database Design

3.4.1 Design and Entity Relationship

The database for MSPIIB is in MySQL which is a built-in relational database
management system. MSPIIB is running on Linux operating system. The database was
designed to support future expansions in the MSPIIB system. Figure 3.1 shows the
relationships among the tables in the database.
3.4.2 Description of Database Tables

This subsection gives a brief description of the tables and fields in the database and their relationships. The following is the list of tables in the database used in the database.

1. Case_details,
2. Procedures,
3. Diagnosis,
4. EMC,
5. Insurance,
6. Login,
7. Procedure_diagnosis,
8. Providers and

**Case_details**

This table consists of fields like `case_no`, `code`, `ins1`, `policy_holder1`, and `group1`. With `case_no` and `code` as the primary key in this table, a user can view the patient case details. A patient may be associated with any number of cases. Patient cases may be added to the case_details table. For a complete list of fields and a description of the table, see Appendix A.

**Procedures**

This table consists of fields like `code`, `description`, `amount` and `ctype`. New CPT codes and descriptions, along with amount of charge for that procedure can be added to this table. `Code` is the primary key for this table. For a complete list of fields and a description of the table, see Appendix A.

**Diagnosis**

This table consists of the fields `code`, `description` and `h_approved`. New diagnosis codes and their descriptions can be added to this table with `code` as a primary key.
**EMC (Electronic Media Claims)**

EMC table consists of fields like *code, name, address, and phone*. This table holds all the details of electronic media claims, and also allows new EMC’s to be added the table. For a complete list of fields and a description of the table see Appendix A.

**Insurance**

This table consists of fields like *name, address, and billing_method*. This table has details of all the insurance companies and accepts new insurance details. For a complete list of fields and a description of the table see Appendix A.

**Login**

The table consists of fields like *userid, name, and password*. Any number of users can be given access to Management System for Patient Information and Insurance Billing. For a complete list of fields and a description of the table see Appendix A.

**Procedure_diagnosis**

This table consists of fields like *case_no, code, and diagnosis*. This table also holds the details of a patient’s case including the procedure and diagnosis. For a complete list of fields and a description of the table see Appendix A.

**Providers**

The provider table consists of all the provider details. The table consists of fields related to the provider like *name, ssn* (Social Security Number), *ftid* (Federal Tax ID) and *address*. For a complete list of fields and a description of the table, see Appendix A.
Patient_details

This table consists of all the patient personal details. This table consists of fields like name, address, and phone_no. For a complete list of fields and a description of the table see Appendix A.

3.4.3 Entity Relationship of Log-in Process

A flow chart of the login process is shown in Figure 3.2. The Home page is connected to the other pages when the user enters the correct userid and password, the PHP script runs for authentication and checks the userid and password in the login table. If the userid and password do not match with the user, an Error page with a login failure message is displayed.

Figure 3.2 Schematic Diagram for Login Process.
3.5 User Interface Design

The Management System for Patient Information and Insurance Billing is secured with login and cookies. Users are authenticated with their specific userid and associated password. After the userid and password are entered they are verified with the user data in the login table. If invalid information is entered, the system responds with an error message. If the userid and password match with the login table the user is directly routed to the main page.

A cookie is a small file that the server embeds on the user's computer. Each time the same computer requests a page with a browser, it will send the cookie too. With PHP, cookie values can be created and retrieved. A cookie will typically contain the name of the domain from which the cookie has been generated. The "lifetime" and the value of the cookie is usually a randomly generated unique number. Session cookies are temporary cookies that remain in the cookie file of your browser until the user leaves the Website.

Management System for Patient Information and Insurance Billing has session cookies for every page, which makes the system more secure and allows a user to carry information across pages of the system to avoid having to re-enter information.

The Transaction module is the main module in this system. In addition to the Transaction module, the system has many small modules like Patients, Diagnosis, Procedures, and Insurance. Each module is briefly described in the following section.

3.5.1 Transaction Module

A short description of the Transaction module function and options available for the input is given below:
**Input** may consist of any of the following: patient code, first name, last name, social security number (ssn), date of birth (dob), none.

**Function:** A schematic diagram of the transaction module is shown in the Figure 3.3. The *transaction.php* script displays the screen with options to enter new transaction, generate claims and generate reports. The *enter-transaction.php* script displays a screen for searching any existing patient. Patient code, first name, last name, social security number or date of birth can be used to do a patient search. In the search results display, each matching patient is provided with *select* or *edit* scripts for further processing. The *case_display.php* script allows the user to view existing matching cases for the patient and create new cases for the existing patient using script *new_case.php*. The *edit_case.php* script allows the user to edit any existing cases for the patient. The *new_case.php* script allows the user to add a new case to the database. Patient insurance information, procedure codes, and diagnosis codes are used in adding a new case to the database. At the end of the day the *generate claims* in *Transaction module* gathers all cases for that day and generates a file which can be sent to the clearing house or EMC service vendor using email. *Generate reports* allows to generate reports.
Figure 3.3 Schematic Diagram for the Transaction Process

3.5.2 Patient Module

The Patient module is another of the major modules in MSPIIB. A description of the module function and the options available for the input are as follows:

**Input** may consist of any of the following: patient code, first name, last name, social security number (ssn), date of birth (dob), none

**Function:** The *patient.php* script displays patient information screen. A text box is provided where identification information like patient code, first name, last name, middle name, social security number (SSN) or date of birth (DOB) can be used to perform a search on existing patients in the database. The *search_details.php* script displays the search results. If the search results display does not contain a matching record for the
patient a new record can be entered into the database using the \textit{new\_personal.php} script. The \textit{edit\_personal.php} and \textit{delete\_personal.php} scripts are also provided to edit an existing patient record or delete it from the database table \textit{patient\_details}.

3.5.3 Diagnosis Module

Description of the module function and the options available for the input are as follows:

\textbf{Input} may consist of any of the following: diagnosis code, description, none.

\textbf{Function:} The \textit{diag.php} script displays diagnosis information screen. A text box is provided where the diagnosis code or diagnosis description can be used to perform a search on the existing diagnosis database. The \textit{search\_details2.php} script displays the search results. If the search result does not contain a matching record for the diagnosis a new diagnosis record can be entered into the database using the \textit{new\_dx.php} script. The \textit{dx\_edit.php} or \textit{dx\_delete.php} scripts are also provided to edit an existing diagnosis record or delete it from the database table \textit{diagnosis}.

3.5.4 Procedures Module

Description of the module function and the options available for the input are as follows:

\textbf{Input} may consist of any of the following: procedure code, description, type, none

\textbf{Function:} The \textit{proc.php} script displays procedures information screen. A text box is provided where procedure code or description can be used to perform a search on existing procedure database. The \textit{search\_details3.php} script displays the search results. If the search result display does not contain a matching record for the procedure a new procedure record can be entered into the database using the \textit{new\_cpt.php} script. The \textit{cpt\_edit.php} and \textit{cpt\_delete.php} scripts are also provided to edit an existing procedure record or delete it from the database table \textit{procedures}.
3.5.5 Electronic Media Claims (EMC) Module

Description of the module function and the options available for the input are as follows:

**Input:** none

**Function:** The `emc.php` script displays options to send electronic claims and EMC receiver record search, addition or deletion.

A text box is provided where EMC code or EMC name can be used to perform a search on existing claims in the database. The `search_details4.php` script displays the search results. If the search result does not contain a matching record for the record a new claim can be entered into the database using the `new_emc.php` script. The `emc_edit.php` and `emc_delete.php` scripts are also provided to edit an existing claim or delete it from the database table `emc`.

Electronic claims will be sent using the `send_claim.php` script. The claim file to be sent is selected using the file browse button. EMC receiver name and email is obtained from `emc` table. Once the user submits the form, an e-mail will be sent using the `mail.php` script to EMC receiver with the claim file as an attachment.

3.5.6 Insurance Module

Description of the module function and the options available for the input are as follows:

**Input** may consist of any of the following: insurance code, name, none

**Function:** The `insurance.php` script displays the insurance screen. A text box is provided where insurance code or insurance name can be used to perform a search on existing insurances in the database. The `search_details4.php` script displays the search results. If the search result display does not contain a matching record, a new insurance can be entered into the database using the `ins_edit.php` and `ins_delete.php` scripts are
also provided to edit an existing insurance company or delete it from the database table Insurance.

3.5.7 Provider Module

Description of the module function and the options available for the input are as follows:

**Input** may consist of any of the following: provider id (PID), first name, last name, middle name, none

**Function:** The `provider.php` script displays provider screen. A text box is provided where provider id or name can be used to perform a search on existing providers in the database. The `search_details5.php` script displays the search results. If the search results display does not contain a matching record, a new insurance can be entered into the database using the `new_provider.php`. The `prov_edit.php` and `prov_delete.php` scripts are also provided to edit and existing provider record or delete it from the database table `providers`.

3.5.8 Safety Module

Description of the module function and the required input are as follows:

**Input:** login name, full name, password

**Function:** The `safety.php` script displays a security setup page with an option to add new users and an option to change password. Only the system administrator has the privilege to add new users to the system. Registered users can change their passwords by providing their username and current password. User name and password are obtained from the `login` table.
3.5.9 Backup Module

Input: password

Function: System backup is done using the backup.php script. Only admin can do a backup. Password is required to start a backup process. These are validated against the login table from database. On successful validation, the database will be backed up onto a file. An appropriate error message is displayed on validation failure.

3.5.10 Exit Module

Description of the module function and the options available for the input are as follows:

Input: none

Function: The Exit module logs out the current user from the system by unsetting the cookie, and redirects the user to the MSPIIB home page.
4. EVALUATION AND RESULTS

Usability evaluation is an analytical study of the usability of a system. The goal of an evaluation is to provide feedback for software development. It is an iterative software development process. Usability evaluation helps designers recognize a problem, understand the problem and its underlying causes in the software, and plan changes to correct the problem [Rosson 2002]. Evaluation of this system included the usability of the different interfaces, module security, learnability, productivity, system recoverability, and user satisfaction.

The Management System for Patient Information and Insurance Billing system was tested and evaluated at Dr Geeta Lele’s Office in Hobbs, NM. Feedback received was used to improve the usability and efficiency of the system. The system was also tested and evaluated at Ramdev Medical Billing Services Inc, Midland, TX, a potential commercial user in the future. The evaluation examined how well the project was implemented and also analyzed the extent to which the stated outcomes of the project were achieved. Evaluation of this system included the usability of the following interfaces:

- MSPIIB home page,
- User authentication,
- Addition and updating of existing records by administrator,
- Electronic Claim Transmittal and System Backup, and
- Daily or monthly report generation,

Additional testing and evaluation was conducted in Computer Science Lab CI 226 on a system with Windows XP operating system, Pentium IV processor with 2.4 GHz.
speed, 256 Mega Bytes RAM and 19 inch monitor. Internet explorer was the default browser used for this testing. MSPIIB was also tested on computers located in Dr. Geeta Lele’s office in Hobbs. Hospital staff and Dr. Lele evaluated the software module and made valuable suggestions for user interface improvement.

The transaction module was tested and evaluated during various stages of system development. Following changes were made to the system:

1. Addition of a link to edit patient personal details directly from transaction module. This prevents the user from toggling between transaction and patient modules to edit personal profile. This feature also improved the ease of software usage.

2. Addition of new patients into the database directly from transaction module. As explained above this change increases the ease of software use and employee productivity.

3. The transaction module was altered to include an option to change existing patient case details rather than creating a new patient case in the event of incorrect case creation. This change minimizes the number of redundant cases for the same transaction and also helps in easy tracking and future reference.

The patient, diagnosis, and procedure modules were tested and evaluated by the provider, hospital nurses and insurance clerk at the Hobbs Office. Feedback was received and a change in the patient module was implemented. Change resulted in the use of first, last, and middle names as a search option in addition to the social security number and date of birth. This provided another option to search patient directory without using sensitive personal information. This change increased the ease of use and personal
information security. Users at both testing locations were really impressed with navigational features of the software system. The flow of the modules was designed to operate in a fashion similar to the sequence a new patient would undergo while visiting a physician. This feature increased the learnability and usage retainability by the users.

Overall, the implementation of the project was largely found to be as planned, with some changes made in response to lessons learned during implementation and from feedback obtained from users. The Management System for Patient Information and Insurance Billing system is currently implemented on a Linux server named penguin at Texas A&M University, Corpus Christi. This system will be installed on computers located in Dr. Geeta Lele’s office. Physicians and Insurance billers are the main beneficiaries of this system.
5. FUTURE WORK

The Management System for Patient Information and Insurance Billing was successfully designed and implemented for healthcare providers like Dr. Geeta Lele in Hobbs, NM. In the future MSPIIB can be upgraded with additional features. A financial module could be added to generate patient statement balances, insurance payment reports, and to generate enough financial data to help the provider file taxes. Each insurance company maintains a unique format for its claims. A feature to format the generated claims specific to a particular insurance company can be added. This feature will reduce the cumbersome process of sending claims to a third party who in turn charge for formatting the claims according to different insurance companies requirements.
6. CONCLUSION

The Management System for Patient Information and Insurance Billing is a well-designed and implemented software module for use by any healthcare provider or insurance biller. It provides an excellent option for storage and easy retrieval of patients’ information. It is also an effective tool to replace the existing paper insurance claims process and manual filing of patient’s information.

MSPIIB was developed using high level programming languages PHP (Hypertext Preprocessor), MySQL and Perl (for reports). Use of PHP, a server-side scripting language designed specifically for the Web, coupled with MySQL, a very fast, robust RDBMS makes MSPIIB more user friendly, secure and robust. The system was developed with utmost care, keeping in view the client requirements and preferences. Consultations with clients and their suggestions at every development phase of the system were a big help in improving the usability of the module. Each phase of the project was reviewed and tested with real time information from clients. The Transactions module provides a glimpse of all transactions for a given patient on a given day. The Patients module acts as an excellent database for all patient information. The Security module keeps the system clean and safe from any unwanted intrusion. The Backup module backs up the database to ensure safety from loss of data.

On the whole MSPIIB saves a lot of time for the staff and insurance billers. In an increasingly technological world, this system can also be used to send data for outsourcing, further reducing the time and expenses for healthcare providers.
BIBLIOGRAPHY AND REFERENCES


APPENDIX A – DATA DICTIONARY

Tables used in the MSPIIB software package are as follows.

**Login**

This table contains the login information of users authorized to use the system with the following fields:

- userid: name used to login into the system
- name: full name of the user
- password: password selected by the user to login into the system

**Patient_details**

This table contains personal information of the patients. This table has the following fields:

- code: a unique number associated with every patient, patient id
- l_name: last name of the patient
- f_name: first name of the patient
- m_name: middle name of the patient
- sex: gender of the patient
- dob: date of birth of the patient
- street: street number and name of patient’s address
- city: city name of patient’s address
- state: state name of patient’s address
- zipcode: zip code of patient’s address
- phone1: patient’s primary phone number
- phone2: patient’s secondary phone number
• marital_status: marital status of the patient
• ssn: social security number of the patient
• assigned_prov: provider allotted to the patient
• sign_on_file: acceptance of the patient
• employer: employer name of the patient
• e_street: street name of patient’s employer
• e_city: city name of patient’s employer
• e_state: state name of patient’s employer
• e_zipcode: state name of patient’s employer
• e_status: patient’s employment status
• w_phone: phone number of patient’s employer
• ext: extension number of patient’s employer
• location: location of patient’s employer

Case_details

This table contains the details of patient’s visits, insurance details, claim status and the amount charged for a visit. This table has the following fields:

• case_no: unique number associated with every visit of the patient
• code: unique number associated with every patient, patient id
• case_date: date of visit to the provider
• doi: date of injury
• nature_accident: nature of accident
• acc_related: whether the injury is related to an accident or not
• state: state name where the accident took place
• ins1: primary insurance carrier of the patient
• policy_holder1: primary policy holder name
• relation1: relationship between primary policy holder and the patient
• policy_no1: primary policy number
• group_no1: group number assigned by the insurance carrier
• assignment1: acceptance of primary insurance
• ins2: secondary insurance carrier of the patient
• policy_holder2: secondary policy holder name
• relation2: relationship between secondary policy holder and the patient
• policy_no2: secondary policy number
• group_no2: group number assigned by the insurance carrier
• assignment2: acceptance of secondary insurance
• ins3: tertiary insurance carrier of the patient
• policy_holder3: tertiary holder name
• relation3: relationship between tertiary policy holder and the patient
• policy_no3: tertiary policy number
• group_no3: group number assigned by the insurance carrier
• assignment3: acceptance of tertiary insurance
• case_charges: fee for a visit
• status: whether the patient claim is sent or pending

Procedure_diagnosis

This table contains the details of diagnosis and treatment procedure for a patient’s visit. This table has the following fields:
• case_no: unique number associated with every visit of the patient
• code: unique number associated with every patient, patient id
• proc: procedure code with which the patient is treated
• diag1: primary diagnosis of the patient
• diag2: secondary diagnosis of the patient
• diag3: tertiary diagnosis of the patient
• diag4: quaternary diagnosis of the patient

Providers

This table contains the details of providers associated with the practice. This table has the following fields:
• pid: unique number associated with a provider, provider id
• name: name of the provider
• credentials: credentials of the provider
• address: address of the provider
• city: of the city name of provider’s address
• zipcode: zip code of the provider’s address
• phone: phone number of the provider
• ssn: social security number
• ftid: federal tax id
• medicare: medicare insurance id
• medicaid: medicaid insurance id
• bcross: bcross insurance id
• commercial: commercial insurance id
• upin: unique physician identification number given by medicare insurance
• emcid: id given by EMC receiver
• license_no: license number of the provider
• sign_file: provider acceptance

Procedures

This table contains the list of procedures used for a patient’s treatment. This table has the following fields:

• code: unique number associated with a procedure, procedure id
• description: description of the procedure
• ctype: code type
• tos: type of service
• pos: place of service
• mod: modifier of the procedure
• amount: amount charged for a procedure

EMC

This table contains the details of Electronic Media Claim (EMC) receivers. This table has the following fields:

• code: unique number associated with a provider, provider id
• name: name of the provider
• email: e-mail address of the EMC receiver
• street: address of the provider
• city: of the city name of provider’s address
• zipcode: zip code of the provider’s address
- phone1: primary phone number
- phone2: secondary phone number
- fax: fax number
- contact: contact person name at the insurance carrier’s office
- comment: notes on EMC
- dataphone: dialup number
- dial_prefix: prefix number to dialup
- dial_suffix: suffix number to dialup
- serial_port: serial port number to dialup
- parity: parity number to dialup
- baud_rate: baud rate of dialup
- data_bits: data bits of dialup
- modem_initialization:
- modem_termination:
- dialing_attempt: number of attempts made before a successful sent
- trans_mode: test or active transmission mode
- submitter_id1: primary submitter id
- submitter_id2: secondary submitter id
- submitter_pwd1: primary submitter password
- submitter_pwd2: secondary submitter password
- pgm_file: file name sent
- extras: extra comments
Diagnosis

This table contains the details of diagnosis. This table has the following fields:

- code: unique number associated with a diagnosis, diagnosis id
- description: description of a diagnosis
- h_approved: whether the diagnosis code is approved by HIPAA

Insurance

This table contains the details of insurance carriers. This table has the following fields:

- code: unique number associated with a insurance, insurance id
- name: name of the insurance carrier
- street: street name of insurance carrier’s address
- city: city name of insurance carrier’s address
- state: state name of insurance carrier’s address
- zipcode: zip code of insurance carrier’s address
- phone: insurance carrier’s primary phone number
- ext: insurance carrier’s phone extension number
- fax: insurance carrier’s fax number
- contact: contact person name at the insurance carrier’s office
- plan: insurance plan name
- type: type of insurance
- procedure code set: selection of procedure code set
- diagnosis code set: selection of diagnosis code set
- pt_sign_file: acceptance of patient on form
- **ins_sign_file**: acceptance of insurance on form
- **phy_sign_file**: acceptance of physician on form
- **print pins on forms**: providers selection of name and pin or only pin on to form
- **billing_method**: type of billing, paper or electronic
- **emc_rec**: EMC receiver name
- **emc_payer_number**: payer number given by emc to provider
- **emc_sub_id**: EMC submitter id
- **emc_ex1**: extra comments
- **emc_ex2**: extra comments