##

File-Mate 1500

Design

Version 4

Written by

Keven Abbott

Tyler Crouse

Kiana Delventhal

Liam Westby

Table of Contents:

1 . . . Cover

2 . . . Table of contents

3 . . . Introduction

4-5 . . . Architecture overview

6 . . . Server design

7-8 . . . Client design

9-15 . . . Database design

16-18 . . . Implementation Plan

#

# **Introduction**

In order to facilitate cooperation between medical offices and insurance companies, the Centers for Medicare and Medicaid Services created the CMS 1500 form. This form, also known as the Health Insurance Claim Form, provides a standardized method of filing and managing claims by medical practices on patients’ health insurance. In conventional paper filing systems, this form can be time consuming to fill out, and require repeated reference of certain data items, especially when working with the same patient. File-Mate 1500, created by Form Magic, Inc., provides a computer-based solution to the form. It allows users to fill out and manage CMS 1500 forms quickly by storing the information for future use. Users can populate fields with existing data, customize printed output, and recall previous forms when needed.

Currently, File-Mate 1500 exists as a single user desktop application. Because all of the data is stored locally, it makes it hard to share information across computers without the need to import and export data for accurate and up-to-date information, leading to a lack of productivity and usability of the system. Working with Dave Schurz, the owner of Form Magic, Inc., our intent is to bring this functionality to the web, allowing multiple users to work off of a single data set from any computer or tablet device, from anywhere they have internet access. To facilitate this, we will use the client-server architectural style as the main template for our system. This will allow patient data to be stored on a centralized server, and be accessed by clients in a distributed fashion. Clients will have the ability to concurrently use and edit this data as long as they are granted the rights to do so.

# **Architecture Overview**



Figure 1

 There are two main components that make up this architecture. The client side is the front end of the software system that the users will be interacting and familiar with. The server side will be responsible for enforcing certain security predicates as well as holding the database which supplies all of the information used to generate the insurance forms. The components interact almost entirely through HTML, PHP and SQL queries. The only architecture that influences this architectural style is the client-server architecture.

## Client Side Overview

:

Figure 2

 Figure 2 depicts the client side. It will supply multiple types of web pages depending on the functionality that the user desires. The web service application side will also deal with assigning a certain user group to a user depending on if they are an administrator, patient, doctor or a trial member. The web application user interface will allow the user to view and alter the patient data. This web application can be viewed through a desktop, laptop or tablet device.

## Server Side Overview



Figure 3

 The server will enforce concurrent and secure access of multiple users, document generation from the desired data that is retrieved and the ability to import and export data to and from the database tables. That same data can be imported and exported to and from the server and the client sides. This data transfer can be seen in Figure 3. The server will be a MySQL server provided by our sponsor.

**Server component:**



Figure 4

Figure 4 represents an overview of the communication among server components as well as possible requests and retrievals of data between them. Using a model view control diagram, the server is comprised of four components: a webpage generator, web services, communication with the database, and a PDF generator. From the webpage generator, entered data and setting changes can be made to the control which will then query the database through the database communication component. From there, data may be returned and then updated through the webpage generator. The webpage generator can also contact the PDF generator to make PDFs upon request.

The database shall only be accessible through user accounts that have permissions to access and manipulate certain data. Administrators will have to option to add users and grant them permissions to the database. Through the use of webforms the user will be able to enter and retrieve data through the use of PHP and MySQL.

Salt along with a user’s password will be hashed before it is stored in the database. Each password is given it’s own unique salt that helps to protect it from an attack. This is used to protect against dictionary and rainbow table attacks.

Forms that are created will be viewable as a PDF through the use of the browser’s built-in PDF viewer or another PDF viewing utility. The user will be able to save and update the document as needed by clicking a button. They will also have the option to save the form as a PDF when it is complete. This will be done using libraries such as FPDF and FPDI.

**Client component**



Figure 5

Since our solution is web-based, the client is expressed as the collection of web pages the user interacts with when using the software. The icons in blue (pages 1-3, 5-6, and 10) represent pages which do not require an active account to access, the pages in green (pages 4, 7-9) require a user-level account, and the pages in orange (pages 11-15) require an administrator-level account.

Page 1 represents the first page viewed by any user upon accessing the File-Mate web application. Pages 2 and 3 represent the process of registering for an account (if that is allowed on the current installation; administrators may choose to disable user registration if they wish to handle the creation of all new accounts) and logging in (existing users skip straight from 1 to 3.) Page 4 is the main page for logged in users which, at minimum, will display a simple message indicating which user is logged in. Page 11 is similar, but for administrators. Page 10 provides information on how to contact Form Magic, Inc.

 Pages 5 and 6 represent a demonstration area, although many administrators will disable this if they also disable registration. In the demo area, prospective users may view a sample account’s page and see an example of the program’s output.

 Pages 4, 6, 7, and 8 comprise the user’s area. Here, the user can create a new CMS 1500 form (Page 7) by filling in patient data; this data is then stored in the database and the form is generated for viewing/printing (Page 9.) Similarly, the user may access a previously filled out form (Page 8) and view/print the final output.

 System administrators will have their own page with useful information (such as number of active users, current system status, and any items requiring their attention (Page 11). The administrator may manage existing users (Page 12) and view their details (Page 14), or create a new user and add them to the system (Page 13). They may also access and modify general settings for the File-Mate web application (Page 15).

**Database component**

 The database will be a MySQL relational database used to manage login information for the users of the website and to store the information that is used to populate the forms. The database will be updated and regulated by the server but the database also places restrictions on the type of data and sizes.

 For the database tables we used MySQL to construct the diagrams. The below is a key for the symbols used in the tables.

 - Foreign key  - Not null

 - Can be null  - Primary key



Figure 12

The above diagram is the overall layout of the database. Users are doctors, nurses and other employees of a medical practice or medical office. Patients go to an office and receive a medical service. CMS 1500 forms are then filled out on their behalf using information about their policyholder, their carrier, provider and the service they received.



Figure 6

The above tables(Figure 6) show relevant information for users. Each user has a unique ID, login name, and email. They also have password which will not be stored in plain text for security reasons. The password will be hashed using SHA512 and the salt. Each login attempt will store a result in the login attempt table with the unique ID of the account, the time of the login and if the login was successful or not where zero is a success and 1 is a failure. Users belong to an office or offices, giving them access to the patients that also belong to those offices. By default users will be placed into a demo office with fake information allowing them to trial the software.



Figure 7

 The above tables(Figure 7) store information about a medical office or practice. Users gain access to patient information through the office to which the patients belong. A patient may not belong to more than one office, meaning there is no interoffice patient sharing.



Figure 9

 The above tables(Figure 9) contain information about insured patients. This information is linked to a patient and used to determine their insured status and information. Primary is used to determine if the policy is the primary, zero for yes and one for no.



Figure 8

 The above tables(Figure 8) show information about the patient and the related tables that use the patients information. Patients receive service from an office, they have insurance and all of this information together is used in part to fill out forms.



Figure 10

 The above tables(Figure 10) hold information about the billing address and other identifying factors of healthcare providers and carriers. These are used to fill out the forms but are not unique to a patient, any patient with Blue Cross will have the same carrier information.



Figure 11

 The above tables(Figure 11) show the total of information that is necessary to fill out a form. The forms are composed of a patient information which includes their insurance status and service they received. The other component of the forms are the provider and carrier information which may not be unique to a specific patient but depends on their coverage.

**Implementation Plan**



Figure 12

 The plan for implementing this system is to have a first effort at each aspect completed within the first month. From here, testing and client feedback will drive the effort to refine and complete the system. Figure 12 demonstrates an example Gantt chart of tasks, although tasks may be completed ahead of time, and reinitiated if new features, or changes in existing functionality, are requested. The darkened areas represent the completion progress of the task as of publication.