DEC10-10 Design Document
Paramedic Information Management System
Mary Greeley Medical Center

Jamin Hitchcock Nathan Dane Eric Tweedt

April 27, 2010

Contents

List of Tables

List of Figures

1 Executive Summary

Our final result will be a web-application that simplifies the job of EMTs at Mary Greeley hospital by allowing them to maintain digital records of their immunizations, certifications, and schedules online through a web based application hosted on one of the Mary Greeley servers.

The primary function of the application will be certification tracking. EMTs are required to maintain certifications for a variety of different medical skills and a lot of these certifications expire after a period of time. Each EMT will get a user account so that he or she can maintain his or her own certifications. The application will allow the EMTs to upload a digital copy of their certifications as they acquire and renew them, they can then view them from any location using a web browser. Each of the EMTs will get their own user account so that they can keep track of their certifications individually. The EMTs will be able to choose whether or not their certifications are publicly viewable or if they are only viewable to themselves and system administrators. The application will also store immunization records, which will be privately viewable to the EMTs and system.

Because many certifications and some immunizations need to be periodically renewed, the system will keep track of the expiration dates of each certification, and will sent the users alerts when one of the certifications or immunizations is about to expire. The alerts will show up on the user.s profile page, and will also be emailed to the users by the application if they have provided an email address.

The final function of the system will be to allow EMTs to keep track of their shifts and submit requests for specific time blocks. Mary Greeley hospital assigns shifts in 8 hour blocks, so the application will have a page that allows a user to select one of the block options and submit it as a request to the administrators. The system will also have a random function that allows for random shift assignment when multiple users request the same time block per request of Mary Greeley.

The system will be implemented mainly in PHP for application programming and MySQL for database implementation. The application will be stored on one of Mary Greeley.s servers and will by administered by Mary Greeley IT. When users upload digital copies of their certifications and immunization records, the copies will also be stored on the server. Users and administrators will be able to log into the server securely to use the system.

The web application project will span from January 2010 until December 2010; development may cease or slow down during the summer months while the developers aren.t enrolled in classes. Costs for this project will include only time put in by each developer, expected hardware costs will include only server maintenance.

2 Audience

This design document is primarily intended to give the senior design staff as well as the client at Mary Greeley Medical Center a detailed summary of the design of the web based information management program. Secondary audiences include other senior design students, computer engineering department staff, and Mary Greeley staff. Other indirect or unintended audiences could also include any public figures not listed here with an interest in the project such as students exploring senior design or anyone who stumbles across the design document.

3 Problem Statement

The paramedics at the Mary Greeley hospital have requested a website to help organize their certifications, immunizations, credentials, meetings, and other important documents. They also would like for the system to send them notifications for when they need to be re-certified, a shift opens, new immunizations are needed, and when yearly TB testing is approaching. All this information also needs to be protected with a log-in system for employees or administrators only

This project will be partially modeled on a product called NinthBrain®. The system will seek to reproduce a small subset of this products functionality for smaller organizations and for a lower price.

4 Operating Environment

Our software is intended to run on the Mary Greeley Medical Center's web servers. If these servers are available we will build our system based on their existing software. If these servers are not available to use, the system will be run on a server provided by the team. The system will run on a Linux, Apache, MySQL and PHP software stack and will be accessible from the Internet.

5 Intended Users

The intended end users of this project will be the paramedics at the Mary Greeley Medical Center. They will not be expected to have a wide knowledge of server systems, computers or even the internet, so the website we need to create needs to be user friendly with an in depth tutorial. As paramedics, they will know what to do or where to go when they need to be recertified, immunized and tested for TB. Our system will simply notify them when these things need to occur and allow them to easily upload documents for meetings, certificates, etc.

6 Intended Uses

The intended use of this project will allow users to upload documents, be notified for certain events, and update their personal profile. Uploaded documents should include certifications, credentials, demographic information, yearly TB testing, immunizations, notes from meetings, and continuing education. Users will be notified for events such as when meetings are going to take place or when a shift has opened up. Other events also include when recertification, immunization, or TB testing is needed to be done. Uploaded documents and notifications are expected to stay along the above lines or of another professional manner. This system should not be able to grant access into any other secure system on the hospital servers or anywhere else. Users should only be able to modify their own profiles, while the administrators will have the right to view and add / delete all users. Uploaded documents should mainly consist of document or pdf files. No uploaded code files will be permitted.

7 Approach

7.1 Objectives

To create a website for the paramedics at the Mary Greeley Medical Center that does the following:

* Organize their certifications, immunizations, credentials, meeting notes, and other important documents.
* Notify users when they need to be re-certified, tested for TB or take new immunizations.
* Track user's work schedules and allow them to request time off and pick up other open shifts.
* Admins need to be able to add/edit all users, allow users to take requested time off and assign open shifts to other users.
* All portions of the website need to be secure from unauthorized users.

7.2 Constraints

7.2.1 Assumptions

It is assumed that all users have a web browser which can render web pages based on current web standards. It is also assumed that the users know how to use said web browsers. It is also assumed that the users are familiar with various web tasks such as navigating a website, using links, filling out and submitting forms.

7.2.2 Limitations

There is a limited amount of time the team has to complete this project. The developers also have a limited knowledge of some web technologies. The developers still have limited knowledge of some web technologies and setting up and maintaining web servers. Doing extra research over the summer will help overcome these limitations.

7.3 Technical Approach

There were two main considerations on the technology for which to create our website on. The first is using Linux, Apache, MySQL, and PHP for the operating system, server, database and website respectively. The second consideration is using Linux, Apache Tomcat, MySQL, and Java for the operating system, server, database and website respectively. The advantages for both of these are that they are all free technologies that can be downloaded online and are fairly easy to learn. The advantage of the first consideration is that PHP is used more widely used in web development and two of the developers are well versed in PHP. The advantage for using the second consideration is that Java is more scalable than PHP, but the disadvantage is that none of the developers have experience in using Java for websites. We chose to go with the first consideration because of the experience the developers have with PHP and the website is being designed for a small organization so scalability should not be an issue.

7.4 Testing Approach

For testing any prototypes and the end product, the developers will be using bugzilla to check the parameters are correct going from page to page and use MySQL queries to see that the information being entered into the database is correct. For further testing, we will be creating a page where the paramedics that will be using the end system will be able to submit any problems or concerns they have with either the functionality or the style of the website. We are hoping to have this page for all prototypes and for the end product to help with upkeep of the site. For first testing the site, the developers will also include a set of instructions to be carried out by the users for testing the system that will end with a form asking if they thought something was wrong or could be improved.

7.5 Recommendations

It is the recommendation of the developers that this project continues as originally envisioned. The project is on course for completion and has a reasonabe deadline with the amount of work left to be done.

8 Requirements

8.1 Functional

8.1.1 Data

The following is a description of the data the system will manage and how that data is related.

User The following information relating to users of the system will need to be saved.

Login Information The system must store username and password pairs for authenticating users. The password must be encrypted to maintain security of the system.

Personal Information The system will store basic personal information including the user's full name, email address, phone number, and address. Multiple phone numbers can be stored for each user.

User Access The system must store information about the level of access granted to each user. Users can be either a regular user or an administrator.

Certification / Immunization For each certification the system must keep track of the name, serial number, expiration date, and a short description of the certification. For immunizations it must store the name, expiration date, and description.

Documents The system must have a mechanism for storing documents. These documents could be either files uploaded to the server or links to external websites. The system will not accept executable files for upload.

Continuing Education The system must provide a way to record time users spend on continuing education. The system should store the name and a short description of the class. The system must also be able to store the amount of time a user has spent on a class.

Scheduling The system should store information for what days and times people are scheduled to work. It should store days and times people want to work, as well as the days and times people do not want to work.

Logs The system should maintain a log of actions users take when using the system. The log should include the user performing the action, the date and time, and a short description of what happened. The log should also record what type of log message it is and what part of the system it is related to.

8.1.2 Behavior

This section describes how the user can view and modify the data managed by the system.

Authentication Only users with a valid username and password will be allowed access to the system. Users can only access the information granted to the user they are logged in as.

User Management An administrator must be able to add and remove users from the system.

Adding Users To add a user the administrator must specify the users name and email address. An email will then be sent to the user with a link to a page where they can set their username and password.

Removing users An administrator can remove a user and all information related to that user. An email will be sent to the user notifying them that they have been removed. When requesting to remove a user the administrator will be asked if they are sure they want to remove the user before the user is removed.

Viewing User Information All users can view their own information. The administrator can view all users' information.

Search An administrator can search for users by name.

Modifying User Information A user can update their own user information. The users full name, email, phone, and address can be updated at any time. The user can update their username at any time, but usernames must be unique. The user can also change their password. An administrator can modify the access level of any user. They cannot modify any other information about a user other than themselves.

Certification / Immunization

Viewing Information All users can view their own certification and immunization information. Administrators can view all users' certification and immunization information.

Modifying Information A user can add and modify their own certification and immunization information regardless of whether they are an administrator or regular user.

Reminders A user can specify how far in advance they should be sent a reminder about an expiring certification or immunization. An administrator can set default times to send reminders, but these can be adjusted by the user.

Reminders will be sent to the user by email when the expiration date is approaching. The user will also be able to view notifications when logged into the system.

Reports Regular users can view a report of upcoming expiration dates for their certifications and immunizations. An administrator can get a report of upcoming expiration dates for all users. Administrators can also view a report of any certifications or immunizations that have expired.

All reports can be given a time frame for which to limit results to.

Schedule

Regular Users Regular users will be able to view the schedule to see when they are scheduled to work and open shifts. A user may request to work during an open shift. A regular user will also be able to request time off for a shift they are scheduled.

Administrators Administrators will be able to see when all people are scheduled to work. Administrators will be able to approve people to work at a specified time. Administrators will be able to approve requests for time off.

Notifications Users may elect to receive notifications about shifts that become open.

Logs

Creating Logs All information saved in the log will be created automatically by the system.

Viewing Logs An administrator may view the system logs. The administrator may filter the logs by user, date, type, system, or any combination of those.

8.2 Non-functional

Easy User Interface The system should be easy to navigate. The display and modification of data should be done in an intuitive way. The layout should be arranged in a way that is familiar to the user.

Compatibility The server side software should be able to run on the current hardware and software Mary Greeley Medical Center already has. Also the website should render properly on the browsers Mary Greeley is currently using.

Availability The system should be accessible from anywhere with an internet connection and web browser. Users should be able to use the system from their homes as well as their work computers.

Security The system should ensure that data is not accessible to people who are not authorized to view it. It should also be designed in such a way that unauthorized parties cannot intercept data they are not authorized to view. The users will also not be allowed to upload executable files which could allow someone to corrupt the system.

9 Functional Decomposition

The systems functionality can be broken up several different ways.

9.1 Subsystems

The system as a whole can be viewed as a collection of several different subsystems. Each subsystems consists of the pages and database tables necessary for viewing and modifying a specific set of information. The system can be broken up into the User, Certification and Immunization, Continuing Education, Schedule, and Document subsystems.

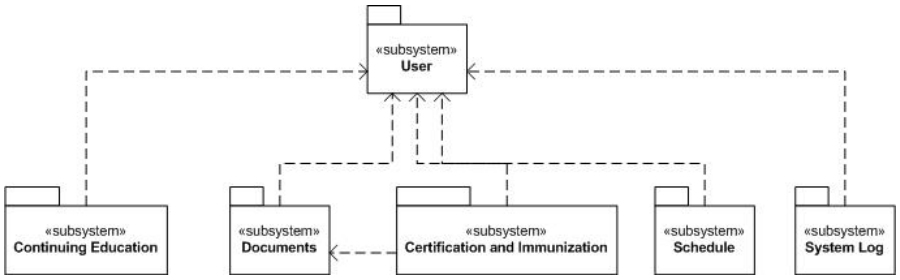

Figure 1: Subsystems

9.2 Functional Blocks

Within each subsystem there are three main functional blocks. These blocks consist of parsing user input, retrieving or modifying database information, and drawing the HTML page.

10 Detailed Design

10.1 Approach

Our group intends to use a rapid prototyping approach to the design of the system. We will try to implement a working version of the system quickly. From there we will be able to write tests for the system to make sure all requirements are met. After the first iteration we will take what we learned from the initial design and improve upon those parts which we feel are weak.

We plan to use the Model, View, Controller design pattern for implementing our system. The design outlined in this document will provide a guideline for development that can hopefully be applied to any MVC framework. We plan to implement the system on multiple frameworks for the initial design including CakePHP, Symfony, and an MVC framework developed on our own. We will evaluate each framework after the first iteration and choose the system which best meets our needs. We will hopefully also be able to apply the best practices we learn from each framework into our final design.

10.2 System Design

The initial design for the system will be based on the Model, View, Controller design pattern. The base system will be implemented based on existing MVC frameworks.

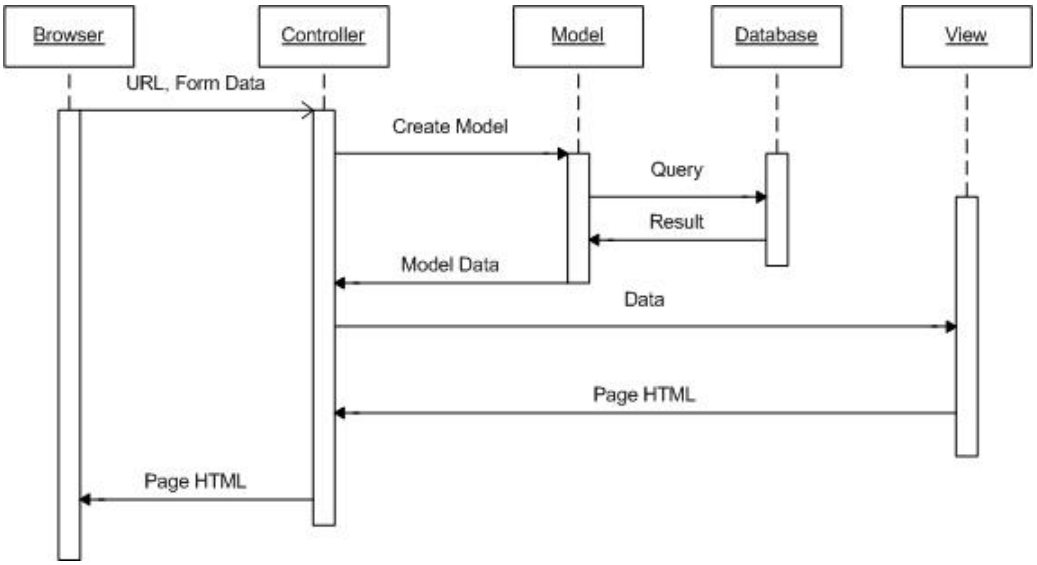

Figure 2: Model View Controller

10.2.1 Models

The model will provide an abstraction layer for accessing the database. Functions for retrieving an entry, modifying and saving an entry, and deleting an entry will be provided by the model class. All business logic will also happen in functions of the model class or subclasses. Functions for returning multiple entries from a table will be implemented as static functions.

User The User model will provide functions for accessing the `user_profile` (Table 2) and `login` (Table 1) tables. This class will also contain functions for checking user credentials for logging into the system. The class will also have functions for retrieving certification and schedule information associated with a given user.

Phone Provides functions for accessing the `phone` (Table 3) table.

Certification Provides functions for accessing entries in the `certification` (Table 4) table. This also provides functions for accessing all certifications that are expired, near expiration, or current. It also has functions for retrieving all documents associated with the certification.

Education Provides functions for accessing entries in the `education` (Table 6) and `user_to_education` (Table 7) tables. This also has functions for returning the total amount of time a user has spent on a given course.

Schedule Provides functions for accessing a single event from the `schedule` (Table 8) table. This also provides functions for accessing all events on a given day, week, and month. Functions for adding users to the `shift_request` (Table 9) table for an open shift event and assigning a user to an open shift are also supplied by this model.

Document Provides functions for accessing a single document from the `document` (Table 10) table. Provides functions for returning all certifications that use the document by referencing the `document_to_certification` (Table 11) table.

SystemLog Provides functions for accessing a single logged event in the `system_log` (Table 12) table. Also provides functions for getting all logged events in the system, clearing old logs, and getting logs for a particular user.

10.2.2 Views

The view will consist of three parts. The first part is a template file that defines the basic layout that is common among all pages. The second part consists of functions that define how each page is drawn from the data provided to it. Finally a set of CSS files defines the style of the pages.

User Views

Admin Main View The Admin Main View (Figure 8) is a simple screen that shows the profile information for a user with admin privileges. This profile will include the available information belonging to the administrator including name, address, phone number, email address, etc. It will also optionally contain a link to the admin's certification page. If the profile is being viewed by the administrator, the "Edit" link will appear at the top of the page. This link is only visible when the user viewing the page is logged in as an administrator. Finally, at the bottom of the page, the "Edit Users" link will appear (only if the logged in as an admin) which takes the administrator to the Edit Users screen where the admin can manage user profiles and certifications.

Admin Edit View The Admin Edit View (Figure 9) is an editor that allows the logged in admin to change his or her profile information and decide whether or not to show the link to his or her certifications page. Checking the box next to the "Show Certifications" link will show it in the Admin Main View page, and un-checking it will remove it from the main view. The "Manage Certifications" link takes the admin to the Manage Certifications page, where he or she can add, update, remove, or otherwise modify certifications belonging to that admin. Selecting the "Manage Users" link will allow the admin to view information belonging to other regular users, including certification information.

User Main View The User Main View (Figure 6) is a simple screen that shows the profile information for a basic user. This profile will include the available information belonging to the user including name, address, phone number, email address, etc. It will also optionally contain a link to the user's certification page. If the profile is being viewed by the user, the "Edit" link will appear at the top of the page. This link is only visible when the user viewing the page is logged in as the user to whom the page belongs.

User Edit View The User Edit View (Figure 7) is an editor that allows the logged in user to change his or her profile information and decide whether or not to show the link to his or her certifications page. Checking the box next to the "Show Certifications" link will show it in the User Main View page, and un-checking it will remove it from the main view. The "Manage Certifications" link takes the user to the Manage Certifications page, where he or she can add, update, remove, or otherwise modify certifications belonging to that user.

Profile Search The Profile Search (Figure 10) screen is displayed when a web user wishes to look up a profile. Searching can be done either by browsing by letter, which brings up a list of all visible user profiles that have a last name beginning with the selected letter, or by typing in a name and getting matching search results. Upon requesting a search or browse, the user will be shown another screen containing a list of matching names, each linking to their respective profile, and a back button.

Certification Views

Certification List View The Certification List View (Figure 11) displays the list of all certifications and immunizations a user has. The page displays two links at the top for adding new certifications or immunizations. Next the page displays a list of all expired certifications or immunizations. Each item in the list will consist of a box with the certification name, the date it expired, a link to update the certification, and a link to any documents associated with the certification. Lists for certifications and immunizations near expiration will be below that, followed by all current certifications and immunizations.

Certification Update View The Certification Update View (Figure 12) displays a form for updating a certification. At the top of the form the certification name will be displayed with the expiration date next to it. Below that the description of the certification will be displayed. Finally two input boxes for entering the renewal date and serial number will be displayed. A button for submitting the form will be displayed at the bottom.

Certification Admin List View The Certification Admin List View (Figure 13) displays a list of all certifications and immunizations tracked by the system. At the top of the page there will be four links: one for viewing expired certifications, one for viewing certifications near expiration, and two for adding new certifications or immunizations. Below the links will be two columns listing the certifications and immunizations in the system. Each item in the list will consist of the name of a certification and a link to edit that certification.

Certification Admin Expired View The Certification Admin Expired View (Figure 14) displays a table showing all users with expired certifications and immunizations. The table will consist of the user's name, the certification that is expired and the date it expired.

Certification Admin Edit View The Certification Admin Edit View (Figure 15) displays a form for adding or editing a certification. First is a text field for entering the name of the certification. Next is a text field for specifying an amount of time the certification lasts before it expires, with a select box for choosing the unit of time. A checkbox for specifying the certifications or immunizations which do not expire will display below the expiration text field. Below the checkbox is another text field and select box for choosing the amount of time before expiration a notification should be sent to the user. Below that is a text box for entering a description of the certification or immunization. If the certification is being edited then a link for managing documents will be displayed below the text box. Finally a button for submitting the form will display at the bottom.

Certification Admin Document Management View The Certification Admin Document Management View (Figure 16) displays a list of documents associated with a certification. At the top of the page is a link for adding a new document. Below that is a list of all documents. Each list item will consist of the document title, a link to the actual document, and a link to remove the document from the certification.

Certification Admin Document Add View The Certification Admin Document Add View (Figure 17) displays a list of documents and a form for creating a new document. The top list will consist of all documents on the server consisting of the document title and the document path. Clicking on the document title will add that document to the certification. Clicking on the path with display the actual document. The upload form will have a text field for the document title, a text field for the document URL, and a file upload field for uploading a document. The add button at the bottom submits the form.

Education Views

Education List View The Education List View (Figure 18) displays the list of all courses a user has completed time in. The page displays a link at the top for adding new courses. Each item in the list will consist of a box with the course name, the total time completed, and a link to update the course.

Education Update View The Education Update View (Figure 19) displays a form for updating a course. At the top of the form the course name will be displayed. Below that the description of the course will be displayed. Finally two input boxes for entering the course date and the amount of time spent on the course will be displayed. A button for submitting the form will be displayed at the bottom.

Education Admin List View The Education Admin List View (Figure 20) displays a list of all courses tracked by the system. At the top of the page there will be a link for adding new courses. Below the links will be a listing of the courses in the system. Each item in the list will consist of the name of a course and a link to edit that course.

Education Admin Edit View The Education Admin Edit View (Figure 21) displays a form for adding or editing a course. First is a text field for entering the name of the course. Below that is a text box for entering a description of the course. Finally a button for submitting the form will display at the bottom.

Schedule Views

User Schedule View This page (Figure 22) includes a list of pending open shifts that the user can try to pick up and the user's schedule. Currently the schedule is listed for one week in normal AM-PM time format. The final result may have an option to view either one week or possibly up to a month and the time will be listed in military time format. Along the side is a simple navigation bar that may switch to an overhead navigation bar later.

Admin Schedule View The admin schedule page (Figure 24) is very similar to the user schedule page with the addition of a list of users requesting time off. Other than that, the admin will be able to try to pick up open shifts and have their schedule listed that same as any other user.

Admin Assign Shift View This page (Figure 26) will be for the admin to assign an open shift to one of the users that requested to pick it up. Date and time will be pre-populated and below that will be a list of all the users that requested to take the shift. Next to each user will be the option to assign them the shift or the admin can choose to assign the shift randomly to one of them.

Request Time Off View This page (Figure 23) will be for any user to request a shift off. The time and day fields may be pre-populated after clicking on the shift in the schedule or can be put in manually. Below that will have a field for the user to describe the reason that the user wants the shift off.

Accept Request View This page (Figure 25) will be for the admin to accept or reject a user.s request for a shift off. All the fields will be pre-populated based on the entry from the database and the admin will decide whether to grant the request or not.

Document Views

Document List View The Document List View (Figure 27) displays a list of all documents on the server. At the top of the page is a form for filtering the results consisting of a text field for title, a date field for the date submitted, and a submit button. Below the form is a link for adding a new document. Below the link is a list of each document. For each document the title and path are displayed with the path linking to the document. Below the name and path are two links for editing and deleting the document. Below those links is a list of all certifications using that document.

Document Edit View The Document Edit View (Figure 28) displays a form for adding and editing documents. The form consists of a text field for the title, and text field for the URL, and an upload field for uploading a document. If the document is being edited and it is a file, there will be a checkbox next to the upload field for the user to specify that the original file should be kept.

System Log Views The Log View (Figure 29) screen shows a table containing a log of past events. In the table new events are added to the top of the table as they occur and are sorted from newest to oldest by default. The page also includes an option to change the sorting so that it sorts events by user, or by the event performed. The logged events will go back as far as they can be found in the log database.

10.2.3 Controllers

The controller interprets the requests sent to it by the browser and loads the appropriate models and views to render the page. The controller is also responsible for processing form data. The controller implements each action as a member function of the controller class. Each action function can accept an array of parameters.

UserController The UserController defines pages for viewing and modifying information about users.

The login page will define a form to be displayed by the view. If the form has been submitted the controller will use the input to load the appropriate User model and call the model's function to validate the users password. If the password is valid the controller will create a PHP session to store the user's authentication information and then redirect the user to the main page.

The logout page will simply log the user out and define a message stating that the user has been logged out.

The user view page will load a User model and pass this information to the view to be displayed. A user id can be specified in the URL to load a specific user's information. The controller will check the access the logged in user has for the specified user to determine which information to load.

The user edit page will define a form with values preloaded from the User model for the logged in user. When the form is submitted the controller will load the submitted values into the model and use the model to save the information to the database.

The user search page will define a form for searching for a specific user. The page will also load all users and organize them by letter. When the form is submitted the submitted search will be given to a User search function to generate a list of users matching the queries. The search list and the alphabetic list will be sent to the view for rendering.

CertificationController The CertificationController defines pages for viewing and modifying certifications and immunizations.

The main regular user page for certifications and immunizations will load all certifications and immunizations for the logged in user.

The regular user certification update page defines a form for users to enter their updated certification or immunization information. The controller will use the id from the user to certification table to load the appropriate certification information.

The main page for administrator certifications and immunizations simply gets a list of all certifications and immunizations.

The expiration list page retrieves a list of all certifications or immunizations that are expired, near expiration, or current. The type of certifications and immunizations to show is specified in the URL.

The certification and immunization edit page defines a form for adding or editing certifications or immunizations. If a certification id is specified in the URL the form will be preloaded with data from the corresponding model. When the form is submitted the submitted values will be added to the model and saved to the database.

The manage documents page for certifications and immunizations will load the certification or immunization specified by the id in the URL and load any documents associated with said certification or immunization.

The add document page will load all the documents on the server. It will also define a form for specifying a new document or web page. If a document id is specified in the URL that document will be added to the certification or immunization specified by the certification id in the URL. If the form is submitted a new document model will be created for that data and the model will be saved to the database. If the user entered both a URL and uploaded a document the controller will report an error. The new document will then be added to the certification.

The remove document page will simply get a certification id and document id from the URL and remove the document from the certification. The user will be redirected back to the manage documents page.

EducationController The EducationController defines pages for managing users continuing education time.

The main regular user page for courses loads all courses the logged in user.

The regular user course update page defines a form for users to enter their updated course information, the date they took the course and the number of hours spent on the course. The controller will use the id from the user to course table to load the appropriate course information.

The main page for administrator courses simply gets a list of all courses.

The course edit page defines a form for adding or editing courses. If a course id is specified in the URL the form will be preloaded with data from the corresponding model. When the form is submitted the submitted values will be added to the model and saved to the database.

ScheduleController The ScheduleController defines pages for viewing and modifying the schedule.

The view schedule page will load all events for the current month for the user as well as any open shifts. The controller will accept an interval parameter for specifying how many days you want to view, for example a day, week, or month. You can also specify a specific date you want to

view by specifying year, month, and day parameters.

The pick up shift controller will get the id of an open shift and add that user to a list of users requesting that shift. The user will then be redirected to the view schedule page.

The time off request controller will accept the id of a shift a user is currently scheduled to work. The page will define a form with a text box for the user to input a reason for the request. When the form is submitted the controller will call a function on the schedule model to save the request.

The admin view schedule will load similar information as the user view schedule except if will load events for all users and all time off requests. The admin controller will accept the same arguments as the user controller.

The admin request controller will get the id of a time off request. The controller will also accept an action parameter, either accept or reject. If the action is accept the request will be changed to an open shift. If the action is reject the request will be changed back to a work shift for the requesting user.

The admin open shift controller will accept the id of an open shift as a parameter. The controller will load all users who have requested this open shift.

The admin assign shift controller will take the id of an open shift as a parameter. The controller also accepts the id of a user requesting this shift, or a special flag specifying a random user should be chosen. If a user id is specified the shift is assigned to them, otherwise the controller randomly chooses a user who has requested the shift and assigns the shift to them.

The admin create open shift controller allows the admin to create open shifts for a given time period.

DocumentController The DocumentController defines pages for uploading and managing documents.

The document view controller will list all documents in the system. The controller will also define a form for filtering the documents by title and creation date. The controller will also load all certifications using each document.

The edit document controller will define a form for adding or editing a document. The form will include a text box for the document title, a text box for a URL, and a file upload field for uploading a document. If a document is being edited and the document is for a file there will be a check box to specify if the original file should be used or if a new file should be uploaded. If the controller is supplied with a document id the information for that document will be preloaded into the form and the document will be updated when the form is submitted. If there is no id specified the form will be blank and a new document will be created when the form is submitted.

The delete document controller gets the document id and deletes the associated document. All entries linking the document to certifications will also be deleted.

LogController The log controller defines a single function for viewing the system logs. A form for filtering the results will be provided.

10.3 URLs

The URLs for accessing the pages in the system will use Apache's `mod_rewrite`. URLs will be expected in the form `http://www.example.com/controller/action/?param=value`. These URLs will be mapped to a request to `index.php?c=controller&a=action&param=value`. The index.php page will use the `c` parameter to create the appropriate controller. The `a` parameter is then passed to the controller to call the action function. The remaining parameters are passed to the action function.

10.4 Cron

A script will be created and scheduled to run periodically using Linux's Cron utility. The script will query use certification model's functions to determine all users that need to be notified of upcoming expirations and send those users emails. The script will also delete old log entries from the system log table.

10.5 Database Configuration

Table 1: login

Field Name	Field Type	Description
id	int	The unique id number
username	varchar(255)	A unique username for each user
password	varchar(255)	A salted hash of the user's password

Table 2: user_profile

Field Name	Field Type	Description
id	int	The unique id number
login_id	int	The id of this user's login credentials from the login table
public	tinyint(1)	Flag for specifying whether user info is viewable by everyone
firstname	varchar(255)	The first name of the user
lastname	varchar(255)	The last name of the user
email	varchar(255)	The email address of the user
address	varchar(255)	The address of the user
city	varchar(255)	The city the user lives in
state	varchar(255)	The two character abbreviation of the state the user lives in
zipcode	varchar(255)	The zip code the user lives in

Table 3: phone

Field Name	Field Type	Description
id	int	The unique id number
user_id	int	The user id from the user_profile table the number belongs to
type	varchar(255)	The type of number this is for example home, work, cell, etc.
number	varchar(255)	The phone number

Table 4: certification

Field Name	Field Type	Description
id	int	The unique id number
name	varchar(255)	The name of the certification
expiration_time	int	The number of time units until the certification expires
expiration_unit	varchar(255)	The units the expiration time is in; days, weeks, months, years
no_expiration	tinyint(1)	Specify whether the certification/immunization has an expiration
notification_time	int	The number of time units until a notification is sent
notification_unit	varchar(255)	The units the notification time is in; days, weeks, months, years
description	text	A description of this certification/immunization

Table 5: user_to_certification

Field Name	Field Type	Description
id	int	The unique id number
user_id	int	The id number from the user table
cert_id	int	The id number from the certification
cert_date	date	The date the certification was acquired/renewed
serial_number	varchar(255)	The serial number associated with the certification

Table 6: education

Field Name	Field Type	Description
id	int	The unique id number
name	varchar(255)	The name of the class or training
description	text	A description of the class or training

Table 7: user_to_education

Field Name	Field Type	Description
id	int	The unique id number
user_id	int	The id of the user who completed time
course_id	int	The id of the course
course_date	datetime	The date the class took place
duration	decimal(5,3)	The amount of time spent on the class

Table 8: schedule

Field Name	Field Type	Description
id	int	The unique id number
event_date	date	The date the event occurs
user_id	int	The person the event is for
shift	varchar(255)	The shift the event corresponds to
type	varchar(255)	The type of event this is: work, time off request, open shift
description	text	A description of the event, used for time off requests

Table 9: shift_request

Field Name	Field Type	Description
id	int	The unique id number
event_id	date	The id of an open shift event
user_id	int	The person requesting the open shift

Table 10: document

Field Name	Field Type	Description
id	int	The unique id number
user_id	int	The user who created this document
title	varchar(255)	The document title
url	varchar(255)	The document url if the document is a website
file	varchar(255)	The path of the file if the document is a file
access	varchar(255)	Who can access this document: public or private

Table 11: document_to_certification

Field Name	Field Type	Description
document_id	int	The id of the document
certification_id	int	The id of the certification the document is associated with

Table 12: system_log

Field Name	Field Type	Description
id	int	The unique id number
user_id	int	The user id of the person logged in performing the logged action
action	varchar	A short description of the action taken
log_date	datetime	The date and time the action was taken

10.6 Testing

10.6.1 Design for Testing

The system will be designed in such a way that it will be easy to test the system. The separation of all business logic to the model classes will facilitate unit testing of that business logic. Database

information and file paths will be defined in a centralized configuration file so that the values can be easily changed to data sources initialized for testing. Error checking will be built into all parts of the system with descriptive error messages to aid debugging.

10.6.2 Unit Testing

We will perform unit tests using the PHP unit testing framework PHPUnit. Unit tests will be written for all the model classes to ensure all the functions are returning the appropriate data.

10.6.3 User Interface Testing

Testing the user interface will be done with Selenium. Selenium can be used to check that pages are being rendered correctly and that all links work correctly. Selenium can also be used to make sure forms can be submitted and that forms report errors when necessary.

10.6.4 Manual Testing

The various parts of the system will be tested manually by the developers. To manually test the system the developer will go through the list of requirements and make sure each task can be completed successfully.

10.6.5 Integration Testing

Once the system is complete it will be installed on different servers to make sure the system can run on machines with slightly different configurations. All automated testing will be run on the different systems to check that everything still works. Some manual testing will also be performed on each new system.

11 Project Milestones

* Project plan completed - Done
* Design documents completed - Done
* Development Environment setup
* Database setup
* Website layout completed
* User/Admin profiles completed
* User/Admin logins completed
* User/Admin file upload capability completed
* Notification system completed
* Website traversal completed
* Writing Tests completed

* Running and Evaluating Tests completed
* Move system to production site
* Technical Documentation completed
* Tutorial / Help site completed
* Poster completed

12 Work Plan

12.1 Effort Estimation

Our project requires is about 31.31 Person Months. This means about 10 months per person on the team. This value was calculated using the COCOMO method. An online calculator was used and can be found at the URL http://sunset.usc.edu/cgi-bin/cocomo81. The following parameters were used in the calculation.

A software product size of 10000 lines of code was chosen. It is difficult to determine the size of a project before it has been created so this number will likely be different than the actual size. A "Semi-detached" software development mode was chosen because not all the team members have a lot of experience in web development.

The project has normal reliability requirements. If the system were to become unavailable for a short time it would most likely only result in delayed notifications and a slight inconvenience for people trying to use the system at that time. The database size should be relatively small because there is not a lot of different data we need to store. Similarly the system is not very complex. Low was chosen for database size and complexity.

Since the project will either be running on Mary Greeley servers or a dedicated server set up by the team computer performance constraints should be fairly easy to meet. Very low was chosen for all computer attributes.

The programmer capability, application experience and programming language experience is normal. Two of the developers currently work as web developers. Analyst capability and virtual machine experience is low because the team is not as familiar with these concepts.

All project attributes are normal. All members of the team have taken classes on modern programming practices and programming tools. The schedule constraints are also reasonable.

12.2 Tasks

Task	Category	Estimated Time	Completed Time
Discuss Requirements	Planning	20 Hrs	18 Hrs
Project Plan Document	Planning / Documentation	20 Hrs	18 Hrs
Research Web Technology	Research	50 Hrs	10 Hrs
Database Design	Research / Implementation	10 Hrs	9 Hrs
Code Design	Research / Implementation	20 Hrs	15 Hrs
Testing Design	Research / Implementation	20 Hrs	5 Hrs
Design Document	Documentation	25 Hrs	20 Hrs
Development Environment Setup	Implementation	10 Hrs	1 Hrs
Program MVC	Implementation	10 Hrs	1 Hrs
CakePHP	Implementation	10 Hrs	0 Hrs
Symphony	Implementation	10 Hrs	0 Hrs
Second Round Programming	Implementation	10 Hrs	0 Hrs
Final Round Programming	Implementation	10 Hrs	0 Hrs
Testing	Testing	50 Hrs	0 Hrs
Bug Fixes	Implementation / Testing	10 Hrs	0 Hrs
Tutorial	Documentation	15 Hrs	0 Hrs
Project Poster	Documentation	5 Hrs	0 Hrs
Weekly Reports	Documentation	10 Hrs	5 Hrs

Table 13: List of tasks and estimated time

12.3 Schedule

ID	Task Name	Start	Finish	Duration
1	Discuss Requirements (Estimated)	2/22/2010	4/16/2010	40d
2	Discuss Requirements (Actual)	2/22/2010	3/11/2010	14d
3	Project Plan Document (Estimated)	2/22/2010	3/3/2010	8d
4	Project Plan Document (Actual)	2/22/2010	3/5/2010	10d
5	Research Web Technology (Estimated)	2/26/2010	4/16/2010	36d
6	Research Web Technology (Actual)	3/10/2010	4/14/2010	26d
7	Database Design (Estimated)	3/3/2010	3/22/2010	14d
8	Database Design (Actual)	3/9/2010	3/30/2010	16d
9	Code Design (Estimated)	3/8/2010	3/26/2010	15d
10	Code Design (Actual)	4/1/2010	4/19/2010	13d
11	Testing Design (Estimated)	3/12/2010	4/8/2010	20d
12	Testing Design (Actual)	4/6/2010	4/23/2010	14d
13	Design Document (Estimated)	3/3/2010	4/23/2010	38d
14	Design Document (Actual)	3/26/2010	4/23/2010	21d
15	Weekly Reports (Estimated)	2/22/2010	4/23/2010	45d
16	Weekly Reports (Actual)	2/22/2010	4/23/2010	45d

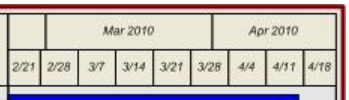

Figure 3: Spring Schedule

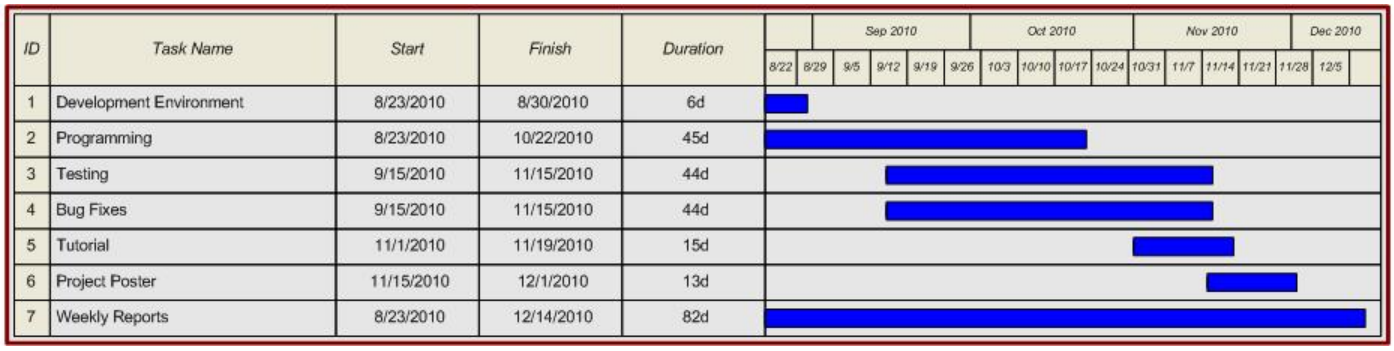

Figure 4: Fall Schedule

12.3.1 Summer Schedule

The summer will be a break for the team and will be used for personal research to increase the developers familiarity with the technologies used to implement the project.

12.4 Resources

Since this project deals exclusively with software that will be run on existing systems the only resources needed will be the engineers' time. This time will be used for planning, research, documentation, implementation, and testing. Mary Greeley already maintains servers so there should be no added cost for server maintenance.

Team Member	Planning	Research	Documentation	Implementation	Testing	Total
Nate Dane	10 Hrs	35 Hr	20 Hrs	20 Hrs	20 Hrs	105 Hrs
Jamin Hitchcock	20 Hrs	30 Hr	20 Hrs	20 Hrs	20 Hrs	110 Hrs
Eric Tweedt	10 Hrs	35 Hr	15 Hrs	20 Hrs	20 Hrs	100 Hrs
Total	40 Hrs	100 Hr	55 Hrs	60 Hrs	60 Hrs	315 Hrs

Table 14: Estimated time per engineer

13 Summary

The design outlined in this document will meet all of the requirements set forth by the client. Upon completion of the project this program will greatly reduce the amount of time spent managing schedules and decrease the likelihood of paramedics failing to renew their certifications on time. With the schedule defined above the design will be implemented before the end of next semester.

Appendices

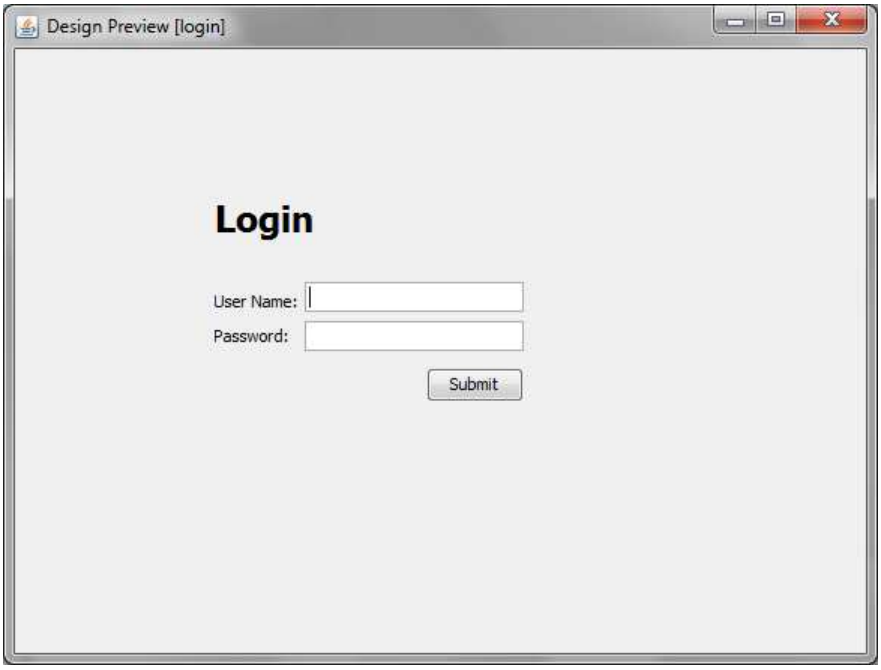

Figure 5: Login Screen

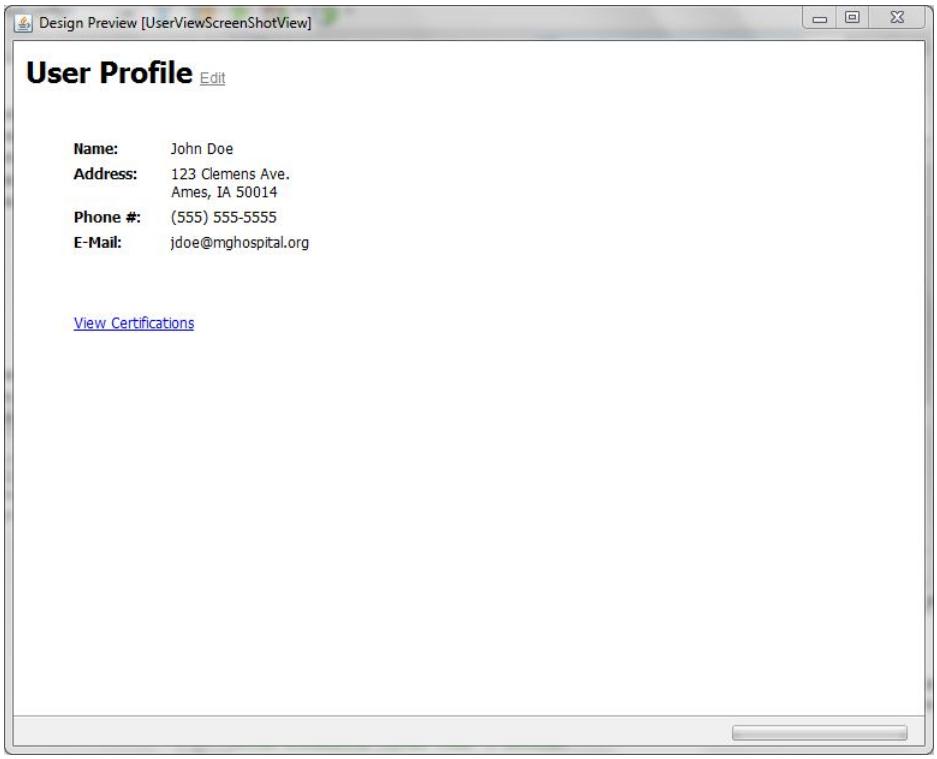

Figure 6: User View Screen

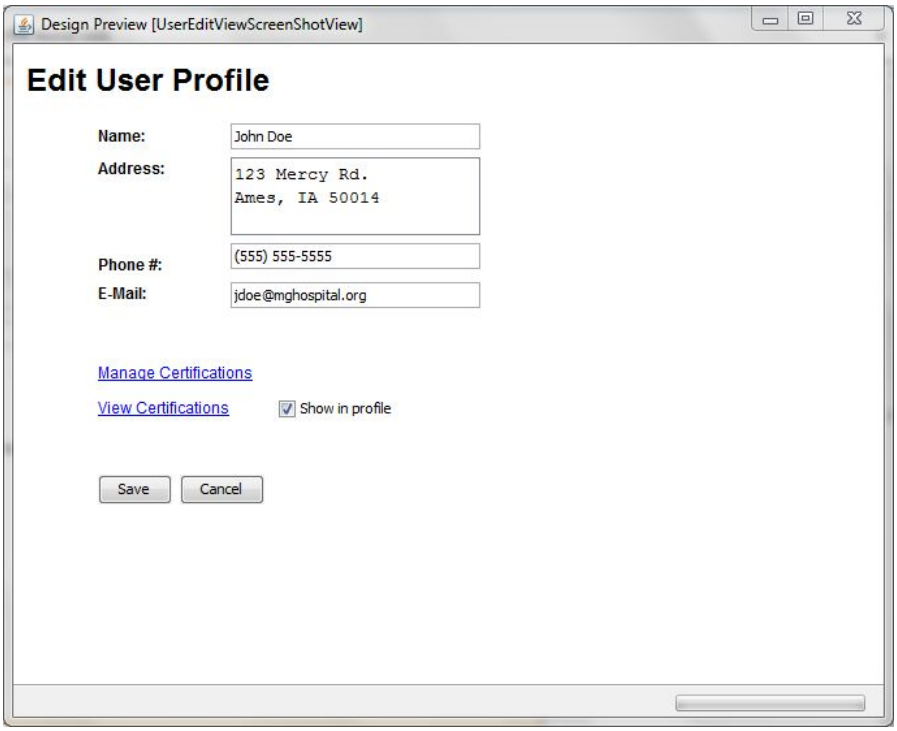

Figure 7: User Edit Screen

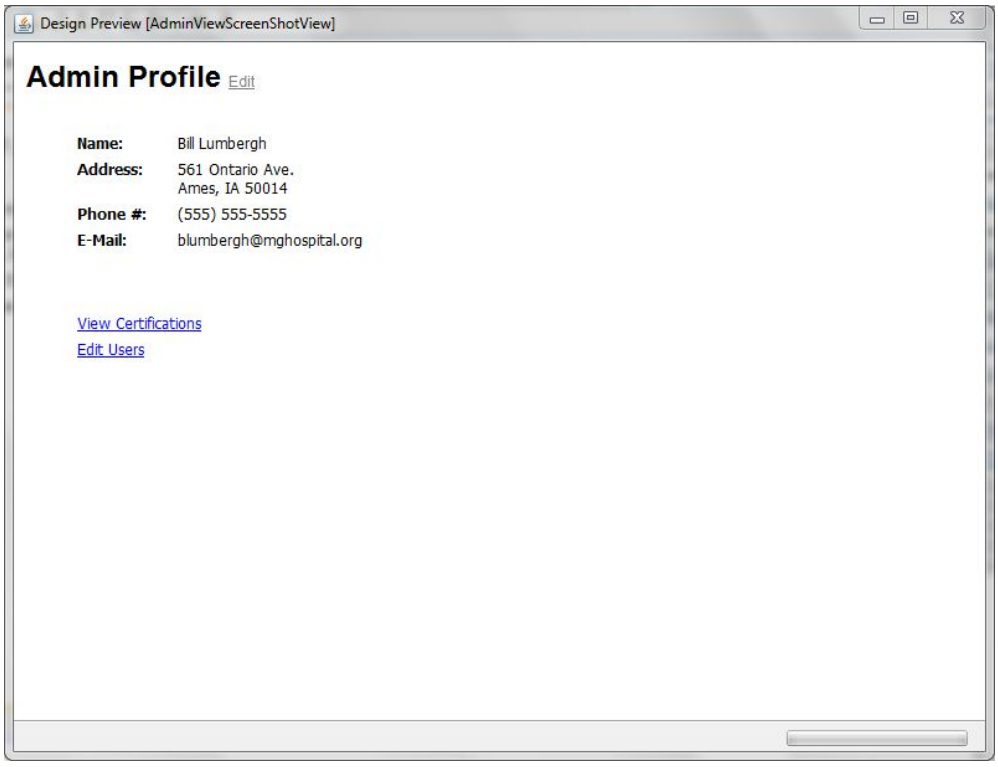

Figure 8: Admin User View Screen

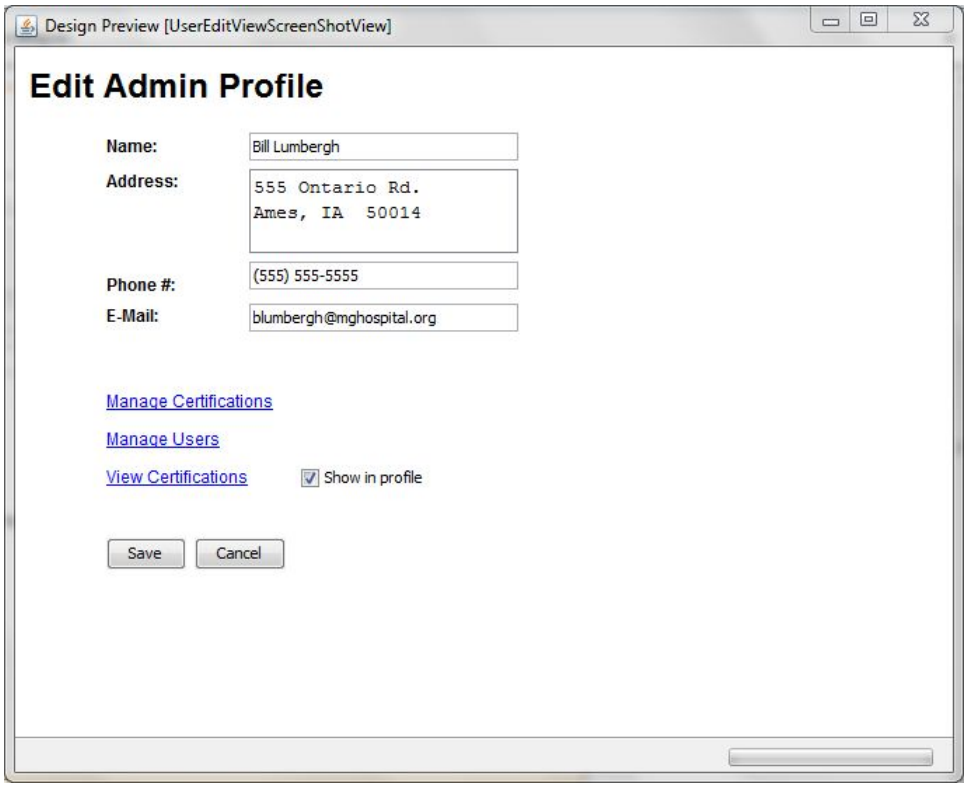

Figure 9: Admin User Edit Screen

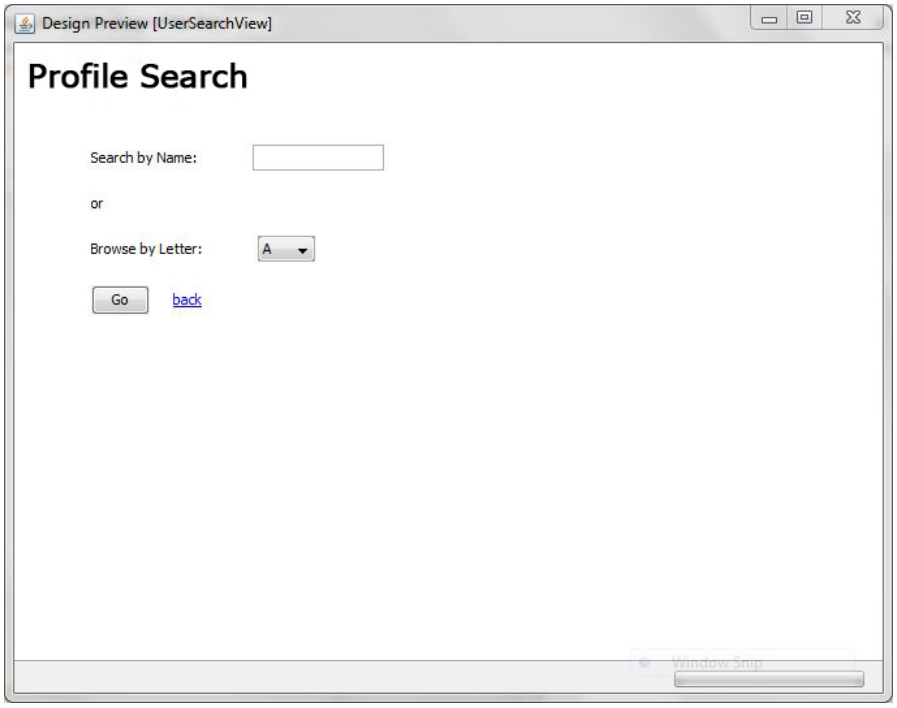

Figure 10: User Search Screen

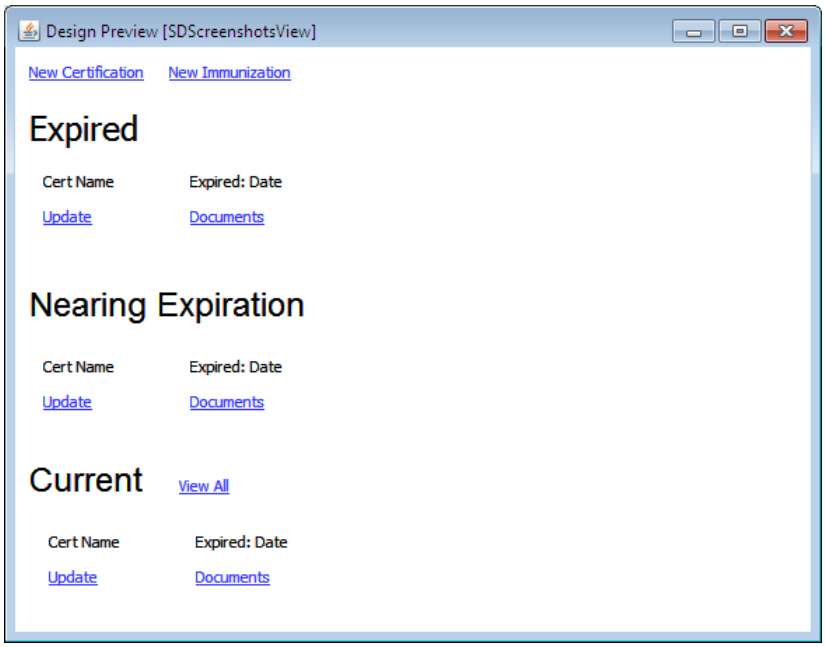

Figure 11: Certification List Screen

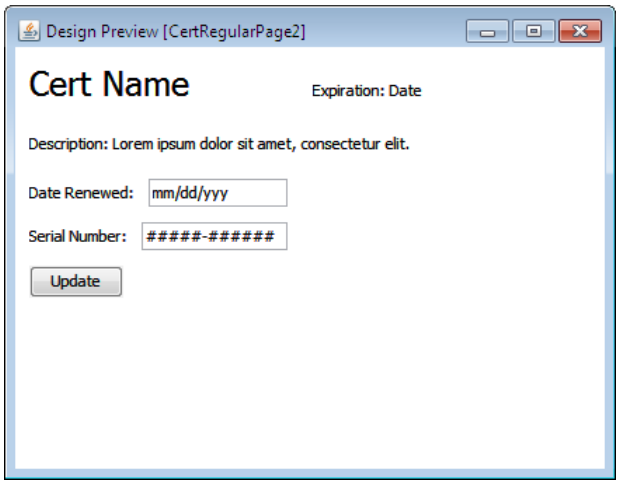

Figure 12: Certification Update Screen

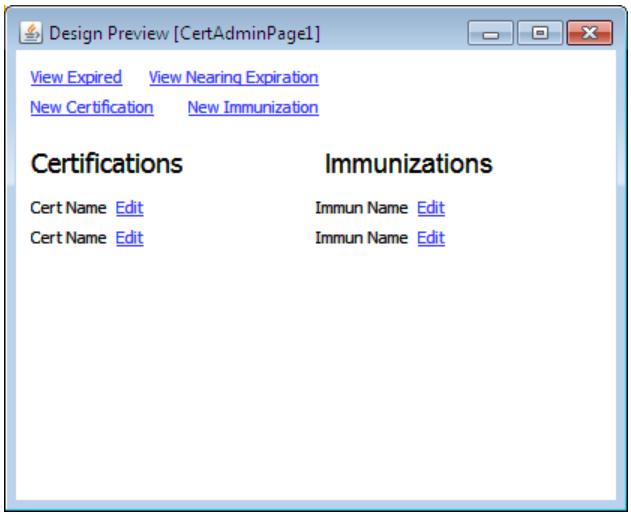

Figure 13: Admin Certification List Screen

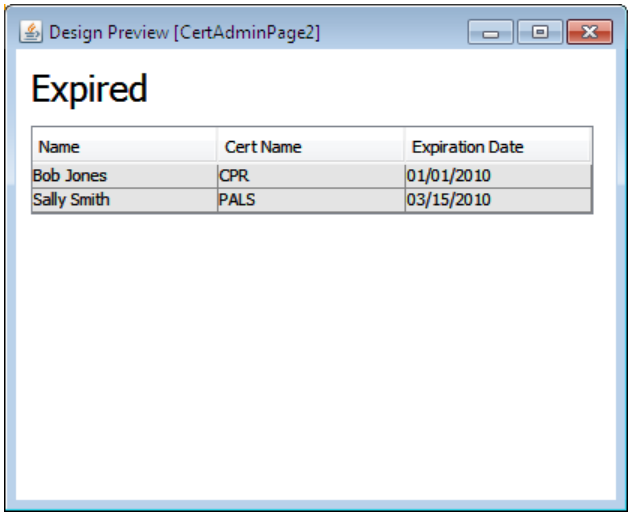

Figure 14: Admin Certification Expired Screen

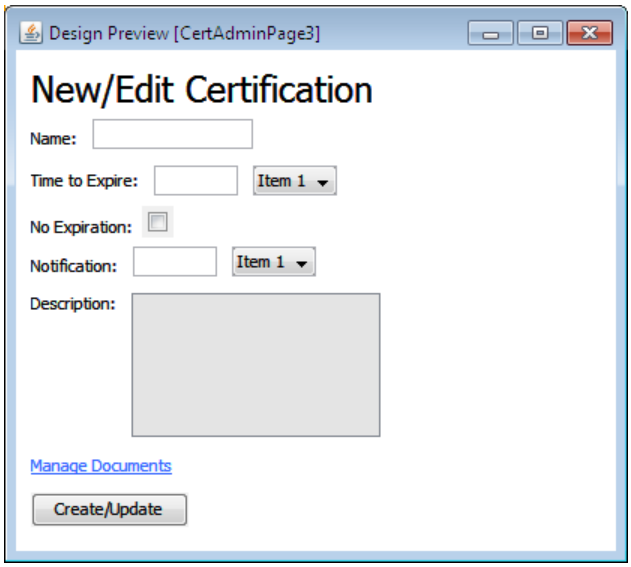

Figure 15: Admin Certification Edit Screen

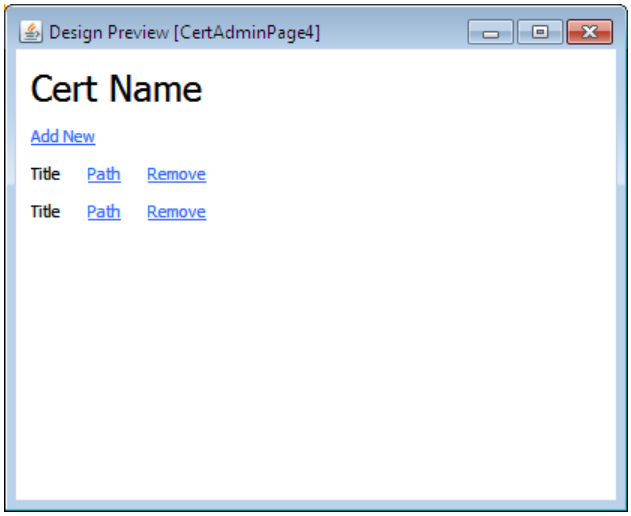

Figure 16: Admin Certification Document Management Screen

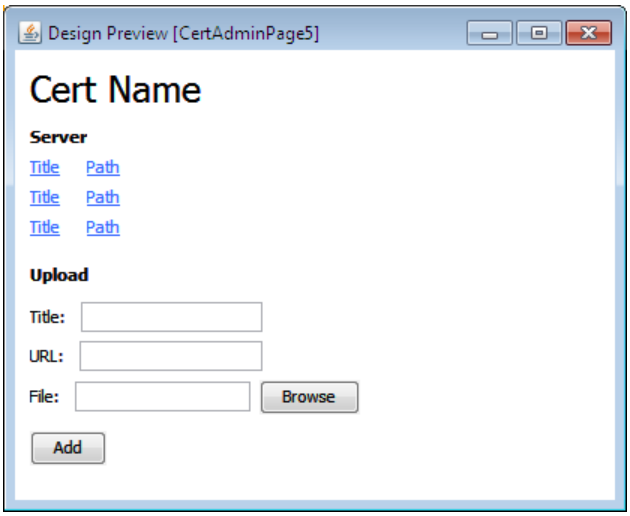

Figure 17: Admin Certification Add Document Screen

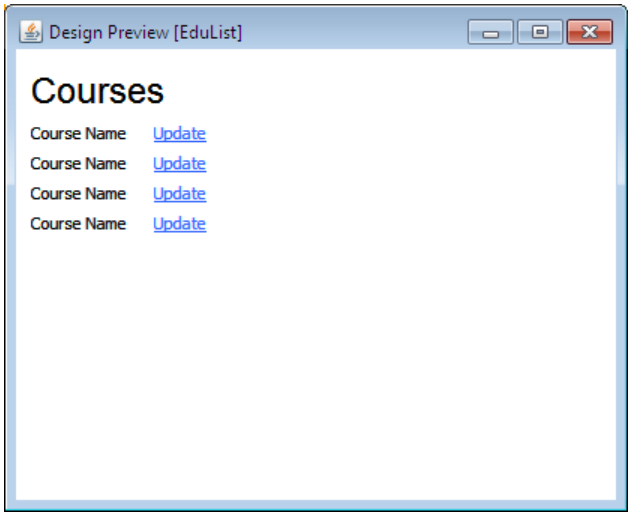

Figure 18: Course List Screen

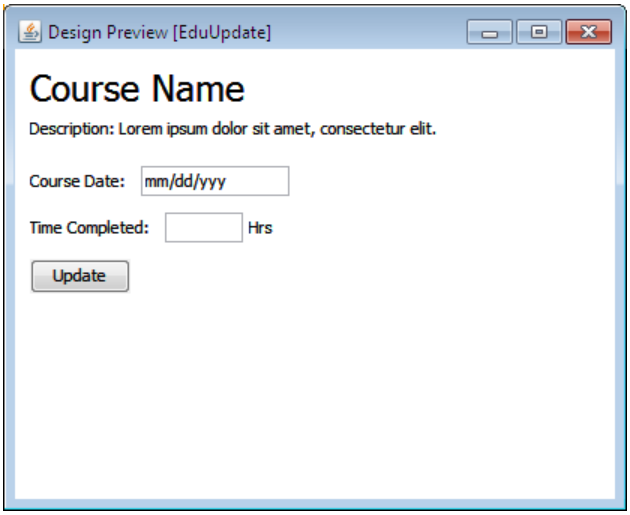

Figure 19: Course Update Screen

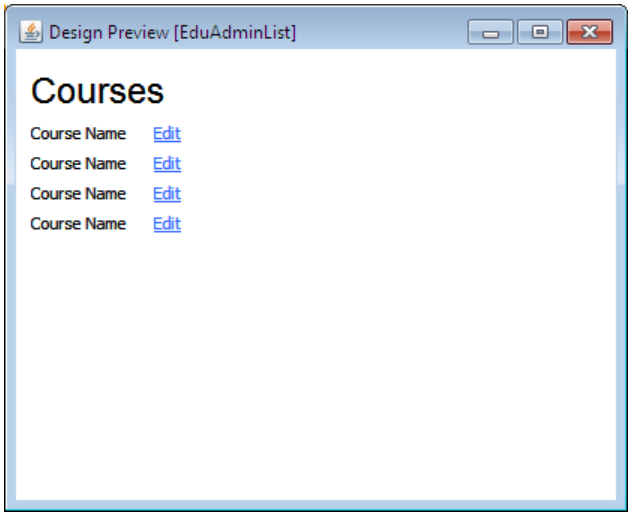

Figure 20: Admin Course List Screen

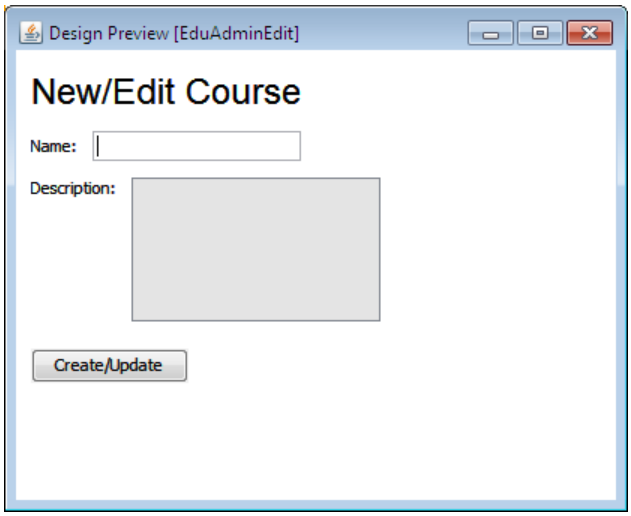

Figure 21: Admin Course Edit Screen

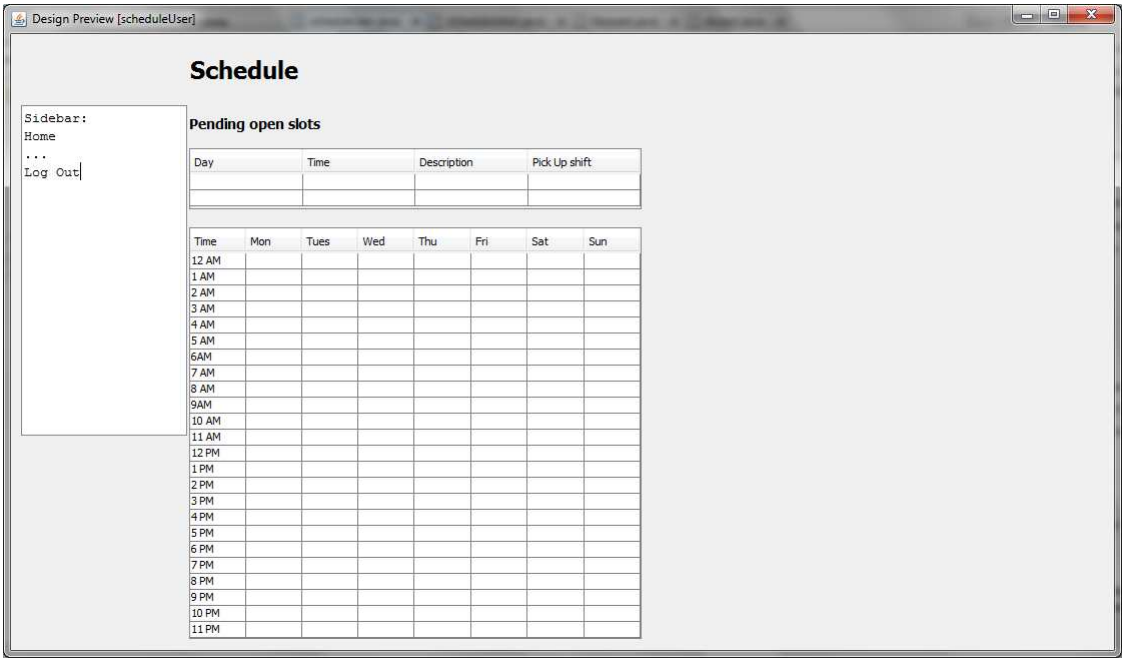

Figure 22: User Schedule Screen

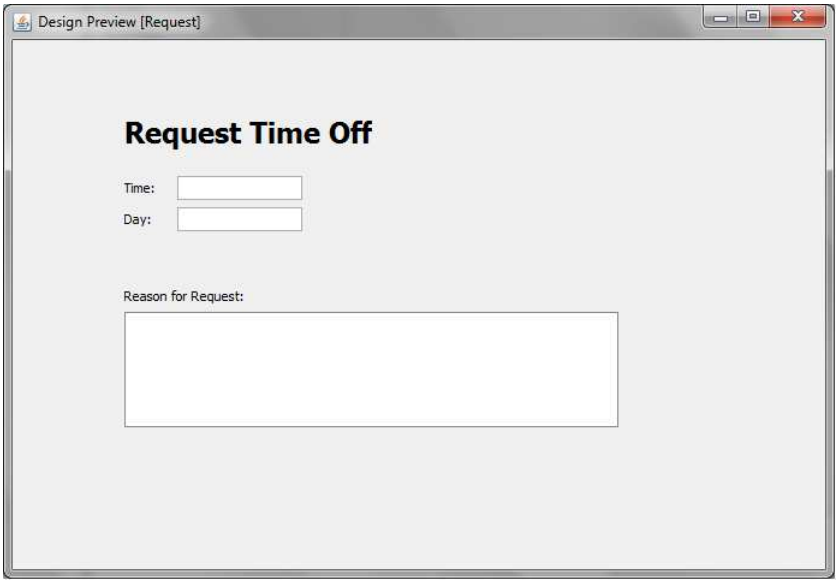

Figure 23: Schedule Request Screen

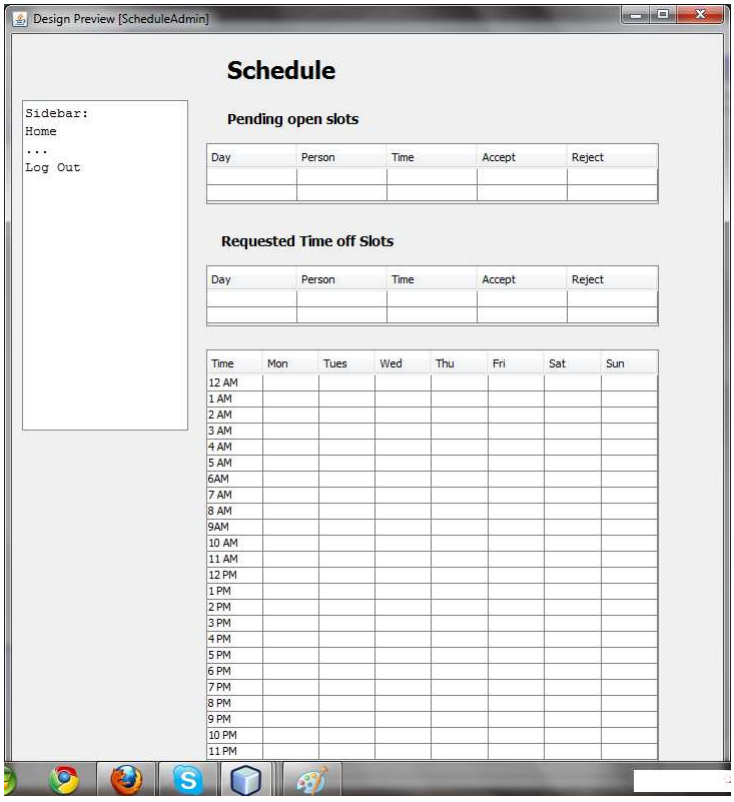

Figure 24: Admin Schedule Screen

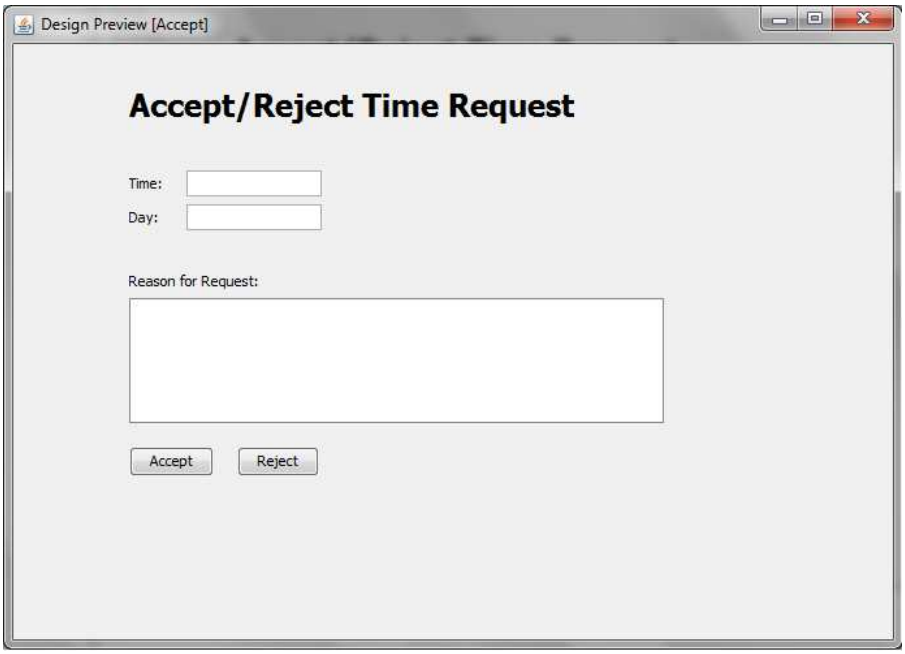

Figure 25: Admin Schedule Accept Screen

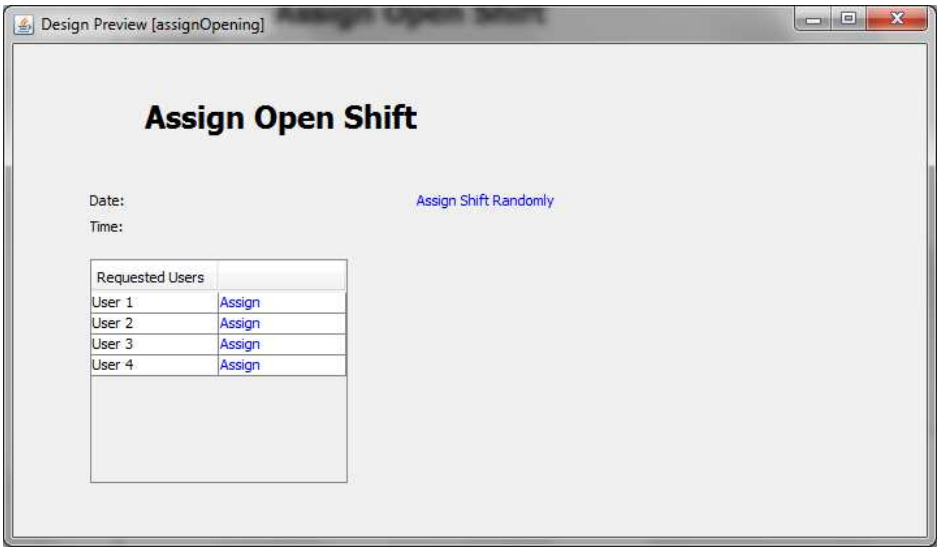

Figure 26: Admin Schedule Assign Screen

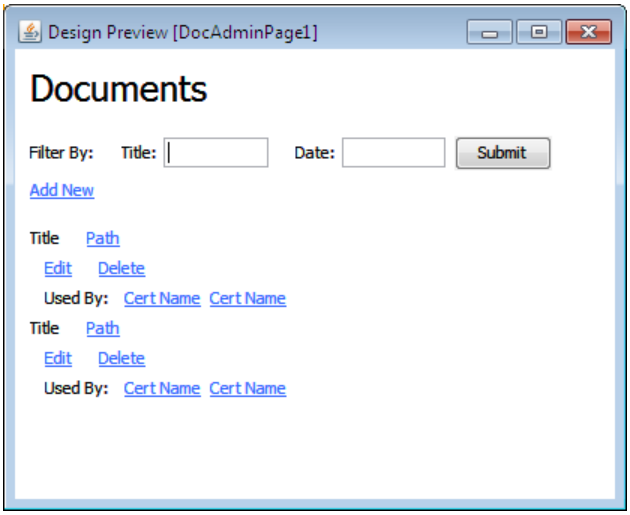

Figure 27: Document List Screen

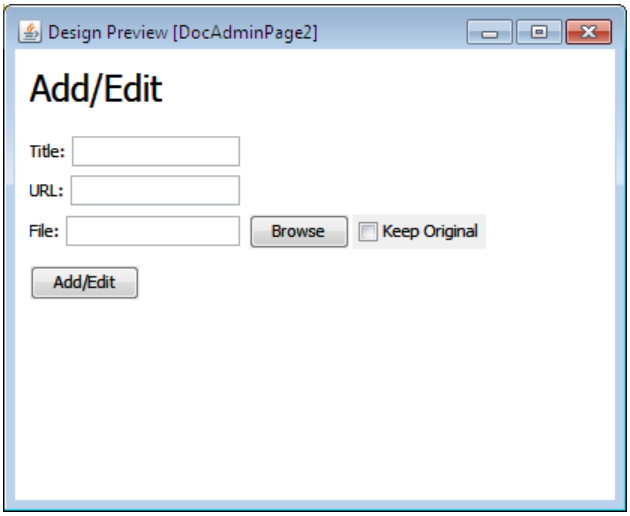

Figure 28: Document Edit Screen

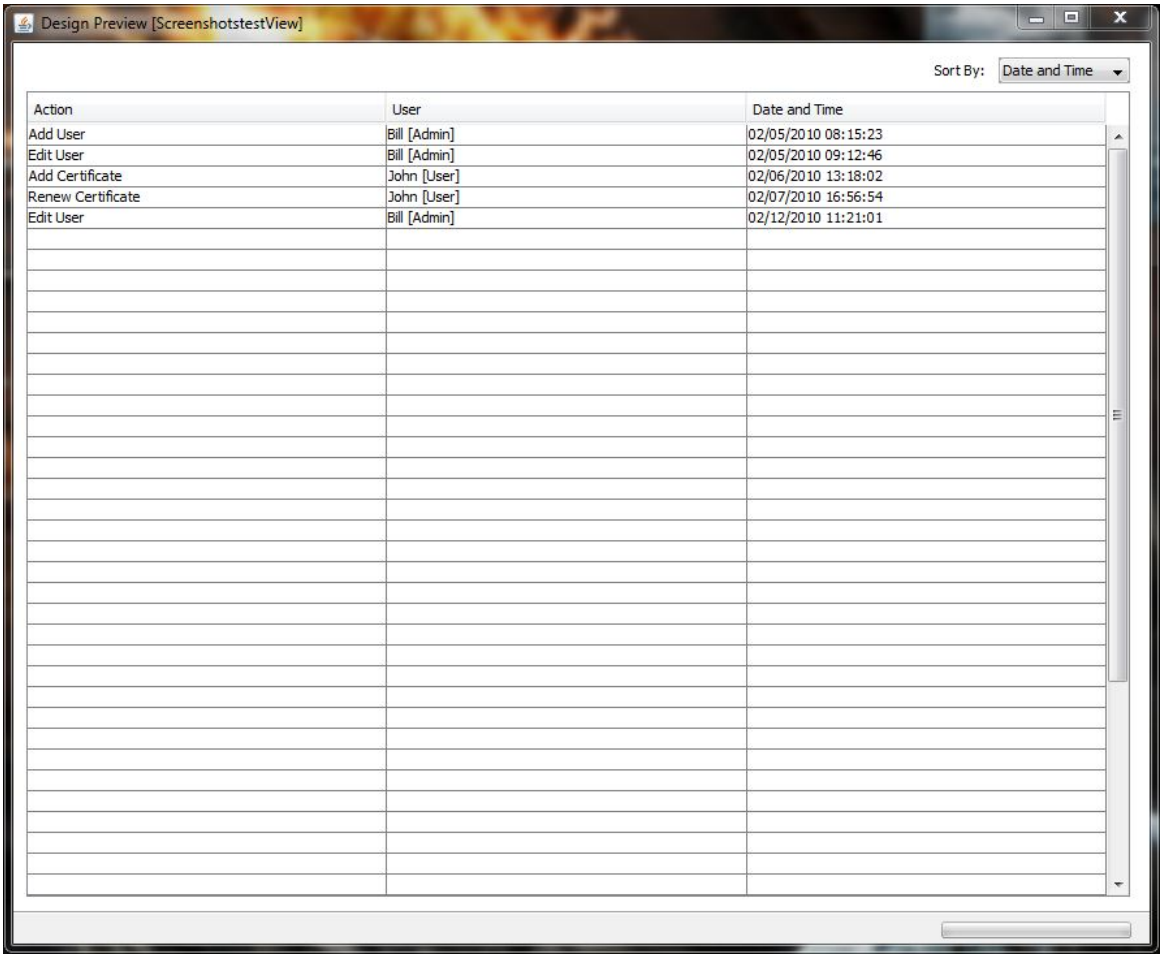

Figure 29: System Log Screen