Computer Science and Engineering, University of Nevada, Reno

ePCR System

Team #27

Kennedy Anukam

Mason Harlan

Yi Jiang

Alec Moore

Devrin Lee

David Feil-Seifer

Bryan Pond

February 26, 2020

Table of Contents

Abstract

The ePCR system is a patient record tracking and entry system designed for use by the San Carlos Rescue ambulance service in Sonora Mexico. The system will allow patient records to be logged and accessed with ease from anywhere through the web. The user interface of the system must be intuitive and support both english and spanish. The database of patient records must be secure. The system could have a big impact on healthcare in lower income regions. This project will be open source in order to benefit as many healthcare providers as possible. This document will outline changes in specification and design of the project.

Recent Project Changes

Since P1 in CS 426, there has been no modification to our project specifications or requirements. Development has been continuing as planned.

Updated Specification

Summary of Changes

Changes have occured in our project's requirements since the previous iteration of the requirements document. One requirement added is the system must be operational on tablet devices for offline use. With offline use, a mobile application must be developed. A requirement that was removed was the need for a timeline of a patient's history. Our project sponsor informed us that a timeline of a patient's history is not needed. Another requirement that was removed was the ability of a user to delete records. Our project sponsor modified the requirement by informing us that charts should be permanent. If changes need to be made to a chart, notes can be added to the chart.

Technical Requirements

Functional Requirements

FR1. [1] - The system shall provide a user interface to complete pick-list sections of a report.
FR2. [1] - The system shall provide a user interface to complete fill in the blank sections of a report.
FR3. [1] - The system shall sanitize user input for the report section to only allow valid entry.
FR4. [1] - The system shall integrate interactive forms that successfully asks for user input necessary for a particular section.
FR5. [1] - The system shall not render information not accessible by an unverified user.
FR6. [1] - The user shall be able to query search for existing patient records.
FR7. [1] - The system shall hash user passwords and store them hashed.
FR8. [1] - The system shall use rate limiting on the login page to prevent bruteforce login attempts.
FR9. [1] - The user shall be allowed to alternate the language of the web application between English and Spanish.
FR10. [1] - The system shall provide an admin mode that will have permission to make changes.
FR11. [1] - The system shall allow the admin to add and delete other users.
FR12. [1] - The system shall lock all records from being edited.
FR13. [1] - The user shall be allowed to add notes to a locked record.
FR14. [1] - The user shall be allowed to add attachments to a record.
FR15. [1] - The user shall be allowed to add interventions to a record.
FR16. [1] - The system shall provide some statistical analysis related to queried features.
FR17. [2] - The system shall allow users to reset their passwords.
FR18. [2] - The system database shall store backup images at a specified interval.
FR19. [2] - The system shall create summary reports for all records.
FR20. [2] - The system shall allow the user batch downloads of patient records.
FR21. [2] - The system shall allow the user the choice to print and or email reports.
FR22. [2] - The system shall allow the user to logout.
FR23. [3] - The system shall provide a multi-factor authentication system that verifies the user trying to sign in.
FR24. [3] - The user shall be allowed to add their own signature to complete authentication, verification, and or correction during record filing.
FR25. [3] - The system shall allow additional organizations to join.
FR26. [3] - The system shall store usage statistics on all users in an organization.
FR27. [3] - The system shall allow a new admin to register a new account.
FR28. [3] - The system shall allow an admin to lock a user account.

Non-Functional Requirements

NFR1.   [1] - All user input sections must have a consistent format.
NFR2.   [1] - Users are able to fill patient records any time throughout the day. A system downtime shall not be needed.
NFR3.   [1] - The system shall interface with a user entirely and securely through a web browser.
NFR4.   [1] - The system shall not allow any access to the database directly, besides itself.
NFR5.   [1] - The system shall be able to operate on various devices.
NFR6.   [1] - A database query shall return a response in an average time of under 10 seconds.
NFR7.   [1] - The system shall have up to at least 100,000 records stored.
NFR8.   [1] - The view charts page shall display up to a maximum number of 5 records per page.
NFR9.   [1] - Users shall be able to access their own user page 95% of the time without failure, of which the 5% should account for site maintenance issues or locked user pages.
NFR10. [1] - The system must not allow invalid URLs.
NFR11. [1] - The system will not allow unauthorized users to access sensitive information.
NFR12. [2] - Users shall receive notification of any updates or additions to the patient records filed by them previously after 2 minutes of the new activity.
NFR13. [3] - The system will support multiple languages.
NFR14. [3] - The system will run on iOS, Android, and all major browsers.

Use Case Modeling

Use Case Diagram

Figure 1 shows the use case diagram for the ePCR system. The actors are shown on the left and the right. Admin users have more functionality than standard users.
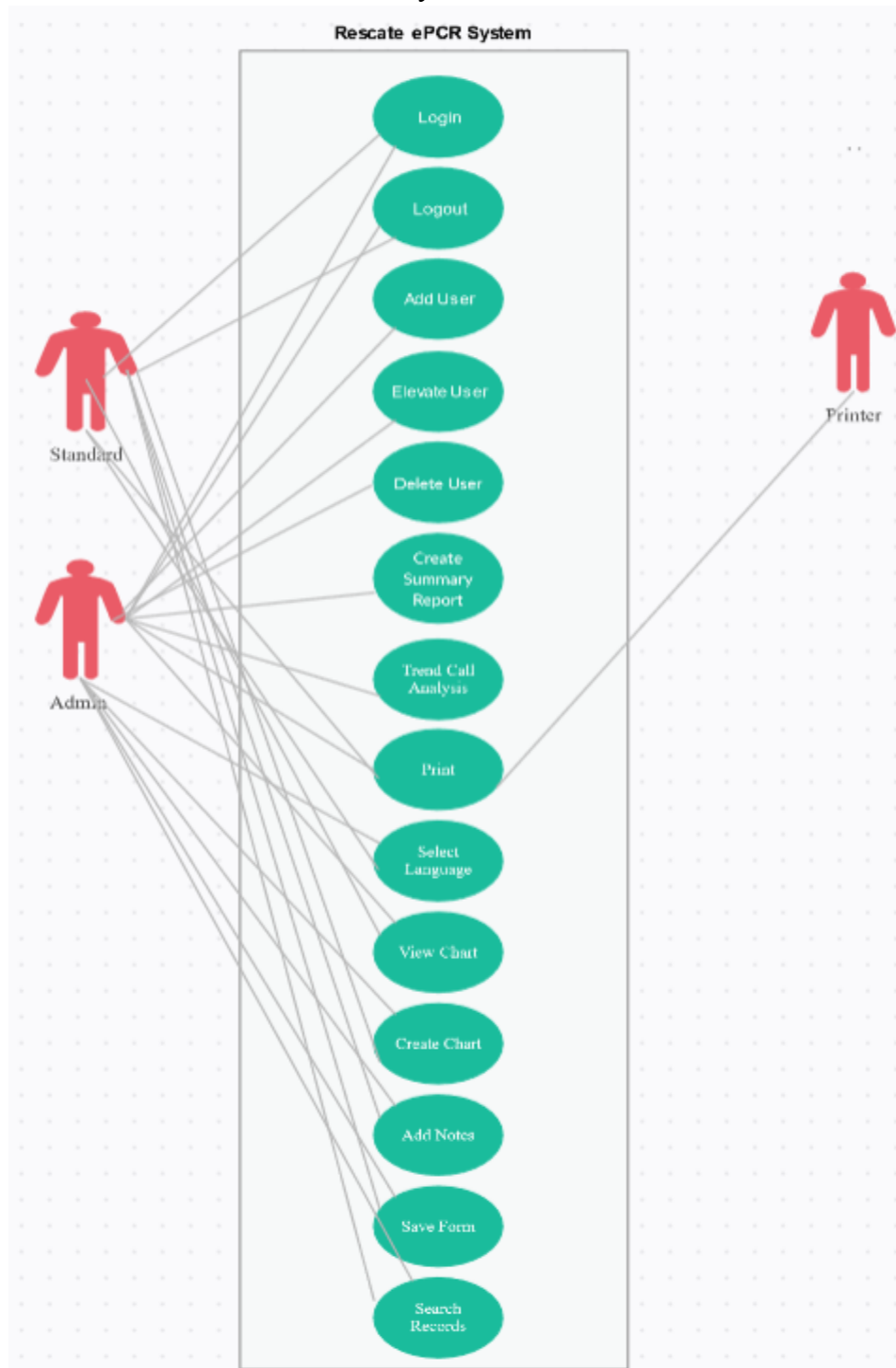

Figure 1: The use case diagram illustrates the main use cases for the Rescate ePCR system.

Use Case Descriptions

Table 1 shows 12 use cases for the Rescate ePCR system. All of the use cases are described briefly.

Table 1: The use cases for the Rescate ePCR system.

ID	Use Case	Description
UC01	Login	The user will enter a username and password on the login page. The system will validate the login and redirect the user to the home page.
UC02	Logout	The system will display a dropdown label to logout. The user will select the logout button to logout. The system will take away the user's permissions. The system will redirect the user to the login page.
UC03	Add User	An admin user will navigate to the settings page. The system will provide an entry box for the user. The user will enter an email of a user to add. The system will display if the addition was successful.
UC04	Elevate User	An admin user will navigate to the settings page. The system will provide an entry box for the user. The user will enter an email of a standard user to elevate to an admin. The system will display if the addition was successful.
UC05	Delete User	An admin user will navigate to the settings page. The system will provide an entry box for the user. The user will enter an email of a standard user to delete. The system will display if the addition was successful.
UC06	Create Summary Report	An admin user will navigate to the statistical analysis section of the dashboard. The user will select two date ranges to create a summary of the charts. The system will return a PDF document.
UC07	Trend Call Analysis	An admin user will navigate to the statistical analysis section of the dashboard. The user will select two date ranges to perform trend call analysis of the calls received. The system will display a histogram of the call volume by time.
UC08	Print	The user will navigate to a chart that they want to print. The user will click a print button. The system will open a print preview window where the user can save or print the chart. If the user clicks save, the system will save the file as a pdf. If the user selects print, the system will send the chart to the printer to print.
UC09	Select Language	The system will provide a drop down of languages to select on the login page. The user will select the language they prefer and the system will update the language throughout the site.
UC10	View Chart	The user will navigate to the view chart section, or use the search function. The user will select the desired chart. The system will return data from the patient records database and display it to the user.

UC11	Create Chart	The user will navigate to the create charts page. The system will display forms to complete the patient chart. The user will fill in the sections for the patient.
UC12	Add Notes	The user will navigate to the patient records report section. The user will select a prompt to add notes to the record. The system will add the user's input to the record.
UC13	Save Form	The user will navigate from the menu to the "create charts" tab. The user will complete the form for the patient with the ability to save the current entered results. The user can go back to worked reports and update the information if not complete. The system will not lock the data until the form is submitted, but the data will be saved to the database.
UC14	Search Records	The user will navigate from the menu to the "reports" tab. The user will be able to search for existing records of a particular patient or look back at the user's submitted reports for any patient associated with the user. The system will render a maximum of five records per page.

Requirement Traceability Matrix

Table 2 shows the relationship between the use cases and the functional requirements. All of the functional requirements are covered by the use cases.

Table 2: The use cases mapped to the functional requirements.

	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8	UC9	UC10	UC11	UC12	UC13	UC14
FR1							X	X		X	X		X	
FR2							X	X		X	X		X	
FR3											X	X	X	
FR4										X	X		X	
FR5	X									X				
FR6							X	X		X				X
FR7	X													
FR8	X													
FR9									X	X				X
FR10	X		X	X	X									
FR11	X		X	X	X									
FR12	X									X				
FR13	X									X		X		
FR14	X									X		X		
FR15	X									X		X		
FR16							X	X						X
FR17	X													
FR18													X	
FR19						X		X						
FR20								X						
FR21						X	X	X		X				
FR22	X	X												
FR23	X													
FR24					X						X	X	X	
FR25	X													
FR26	X													
FR27	X													
FR28	X													

Updated Design

Summary of Changes

There are various changes that occurred in the project's design compared to the previous Design document. Aside from how the system is set up, the components remained the same, but only the inner workings of each component have gone through some advancements. The team has made several updates to the database tables, which is the basis for storing the information required for each chart form. Originally, the team planned to use delimiters for some field columns of the "chart" tables due to the voluminous amount of input fields during reporting. Several input fields have been added to the form on the frontend as well as the database tables. There were some individual columns that stored its own information to help the trend call analysis. In the analysis, the team has to make a summary of the trends for all the reports filed into the system, so there were some variables that needed to be sorted out easily. Methods are still in development to help make the chart creation process more efficient, as most of the fields are reused in several functional requirements.

In addition, while discussing with our sponsor, the team also has a clear direction of what can be changed for the patient and its link to each chart. Our sponsor noted that the system should be able to search for existing patients and autofill their information on the chart or let the user fill in new patient information should the patient not already exist. This also brings about changes to the "patients" table as we initially updated it with other variables such as Braslow color, weight, address, phone, and history. According to our sponsor, these were not necessary for the "patients" table as the sole purpose for said table is to retrieve existing patient information or to add a new patient to it. As such, the team had to make these changes to make the chart creation understandable to our sponsor's team. Finally, the "notes" and "procedures" tables have been easily adjusted to ease the storing and rendering of information to each chart. These were needed to be implemented to add to existing charts because our sponsor required that the charts could not be updated afterwards.

High-level and Medium-level Design

System-Level Diagram

The context diagram in Fig. 2 illustrates the high level components of the system. The Rescate ePCR system is connected to seven other systems. The report system is for creating and storing patient records. The query system is for fetching stored patient records. The security system is to keep unauthorized users out and maintain data integrity. The statistics system is for calculating measurables based on the reports in the system. The ePCR system is connected to a report generation system to create summaries based on the patient data. The renderer system is for rendering the web application based on the user. The ePCR system will have multiple users and will be rendered based on the user utilizing the system. The ePCR system will be connected to the admin system to add, delete, and manage users.
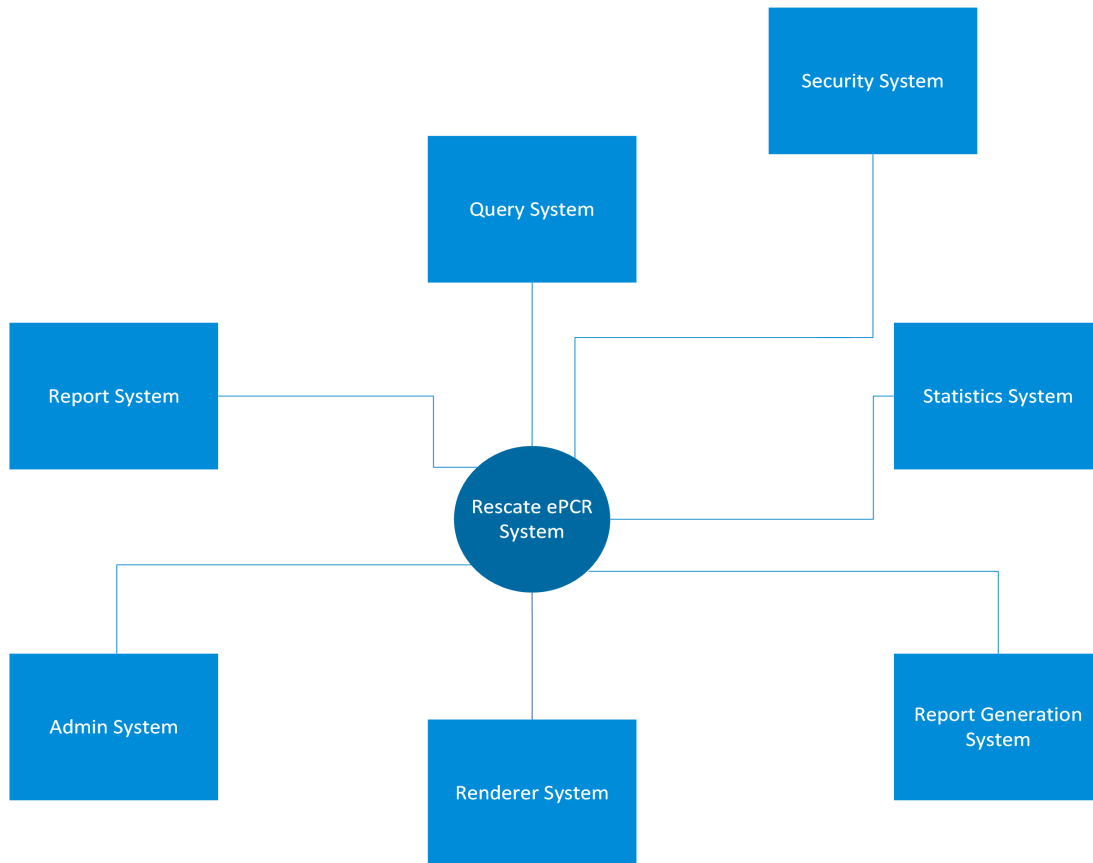
Figure 2: The context model of the Rescate ePCR system. The system interacts with seven systems to complete its functionalities.

Program Units

The Service uses a separated MVC (model, view, controller) framework. The frontend contains the views, while the backend contains the models and controllers.

The backend utilizes a REST API for processing http requests and responses. Each step in backend routing follows the same format:

Router
      Which names routes and passes them to *Controller*
      Input: an http request
      Output: an http response from *Controller* to requester
Controller
      Which reads requests and sends responses
      Input: an http request from *Router*.
      Output: an http response to *Router*
Repository
      Which contains the heavy code to access the database, etc.
      Input: selected items from request body
      Output: a database entry, or success message upon an operation.

<u>Hierarchical View</u>
    Frontend
        React.JS
            App
            Uses Router
                Views
                    Dashboard
                    Login
                    Settings
                Components
                    User
                        Admin Settings
                        User Details
                        User Settings
                        Register
                    Chart
                        View Chart
                        Add Call
                        Add Assessment
                        Add Vitals
                        Add Interventions
                    Patient
                        Add Patient
                        View Patient
                        View Patients
           Router
          Verifies user has access to a protected route
                Authorization
                Handles authorization and stores temporary information for the user required by the app.

    Backend
        REST API
        Uses Server Middleware and Database Handler
            Main Router
            Which accesses all sub routers
            Controller
                Register
                Login
            Repository
            /Users
                Router
                Which names routes under the /users/ url prefix and passes them to Controller
                Controller
                Repository
            /Patients
                Router
                Controller
                Repository

/Charts

PDF Creator
Which creates a PDF from the Database Charts Model

Router
Which names routes under the /patients/ url prefix and passes them to Controller

Controller

Repository
Which can call PDF Creator if requested

/Notes

Router
Which names routes under the /notes/ url prefix and passes them to Controller

Controller

Repository

Server Middleware

Authorization

Uses Database Handler
Which handles the JSON web token verification and creation, and blocks unauthorized access

Bruteforce Rate Limiting
Uses Database Handler
Which only allows a certain number of login attempts per IP address over a period of 2 hours.

Database Handler
Which connects to the SQL database and passes the connection to the other parts of the backend

Charts Model
See 'Database Tables' section

User Model
See 'Database Tables' section

Patient Model
See 'Database Tables' section

Procedures Model
See 'Database Tables' section

Notes Model
See 'Database Tables' section

Database Tables

The below tables are what the team has been working with. The team has discussed for any changes with the external advisor, and some changes are requested to be made.

Patients

Table 3: The "patients" table contains fields that store information about the patient. There are some fields that have been requested to be taken out by the external advisor. The team is still working on the original table for testing purposes.

Field column	Field description
id (primary key)	ID of patient added in the database.
fname	The first name of the patient (separated from lname for search purposes).
lname	The last name of the patient (separated from fname for search purposes).
birth	The date of birth of the patient.
gender	The gender/sex of the patient (necessary for trend analysis).
weight	The weight measurement in kg of the patient. - Requested to be placed in the "charts" table instead.
classify	The classification whether the patient is an adult, child, etc. (necessary for trend analysis). - Requested to be placed in the "charts" table instead.
bcolor	The Braslow Color of the patient. - Requested to be placed in the "charts" table instead.
address	The address of the patient. - Requested to be placed in the "charts" table instead.
phone	The phone number of the patient. - Requested to be placed in the "charts" table instead.
history	Short notes on patient medical history. - Requested to be placed in the "charts" table instead.

Users

Table 4: The "users" table stores information about the crew member.

Field column	Field description
id (primary key)	ID of the user added in the database.
username	The username used by the user to login to the system.
password	The password used by the user to login to the system.
email	The email of the user registered in the system.
phone	The phone number of the user.
name	The real and full name of the user.
privilege	What type of privileges the user has, i.e. "admin" or "standard".

Charts

Table 5: The "charts" table updated with fields that can be easily summarized and sorted in the trend reports. There are some fields that contain delimiters to some inputs from the chart form. The team is working on an efficient method to store and retrieve such data. In addition, there are other fields that have yet to be created for other sections of the form.

Field column	Field description
id (primary key)	ID of the chart added in the database.
date	The date the incident occurred.
incident	Description and details of the incident.
location	The location type of the incident (necessary for trend analysis).
nature	The nature of the call (necessary for trend analysis).
disposition	The disposition type, i.e. "treat and release", "transport", "DOA", "AMA" (necessary for trend analysis).
destination	The destination for the patient transport, i.e. "Rescate Clinic".
agency	The agencies involved, i.e. police, fire brigade, on scene.
trauma	The cause of trauma (necessary for trend analysis).
mci	Checked to fill in the number of patients and Triage Color.
va	Checked to fill in vehicle accident information: type, impact, safety equipment, speed, ejection.
vitals	List of vital test IDs found to be in use from the vitals table.
medications	List of medication IDs found to be in use from the medications table.
procedures	List of procedure IDs found to be in use from the procedures table.
times	Response times: dispatch, enroute, arrive scene, patient contact, depart scene, etc.
patientID	Uses an existing patient ID from or creates a new one in the database.
userID	Any crew member user ID associated with the call.

Procedures

Table 6: The "procedures" table contains fields that are all set to NULL since some procedures do not require it. In the field description, the bulleted points show the specific procedures that use that particular field column. The team will use the field "name" variable to determine which fields the system should render out for the specific procedure performed.

Field column	Field description
id (primary key)	ID of procedure added in the database.
name	Name of procedure to be performed.
time	Time the procedure was performed on the patient. - All procedures
location	Location of body where the procedure was performed. - Hemorrhage Control - Splinting - Suction - Pleural Decompression - IV - IO IV
type	Specific type of procedure performed for the selected procedure. - Hemorrhage Control - Splinting - Spinal Precautions
size	A measurement used in specific procedures. - IV - IO IV - Advanced Airway - LMA
tube	The tube size measurement used in specific procedures. - Advanced Airway - Intubation - Cricothyrotomy
needle	The needle size measurement used in specific procedures. - Pleural Decompression - Cricothyrotomy
fluid	Fluid used for the specific procedure performed. - IV - IO IV
result	- Blood Glucose
delivery	- Oxygen
amount	- Oxygen
adjuncts	- Basic Airway - BVM
physician	- MD consult
orders	- MD consult
teeth	The depth at teeth measurement.

	- Advanced Airway - Intubation
confirm	- Advanced Airway - Intubation
findings	- 12 Lead EKG
rhythm	- 12 Lead EKG
mode	- Cardiac Pacing
rate	- Cardiac Pacing
output	The mA output result. - Cardiac Pacing
capture	- Cardiac Pacing
patientID	Who was the patient that received the procedure.
userID	Who performed the procedure.

Notes

Table 7: The "notes" table stores information of notes that can be added to the chart. This is done so that the original parts of the chart cannot be edited and any summary of changes are made here by the users.

Field column	Field description
id	ID of chart notes added in the database.
body	The text of the note.
chartID	The chart which this note belongs to.
patientID	Which patient the notes are for.
userID	Who wrote the note.

Hardware Design

Our project is not a hardware project. The system only requires a device that can access the internet.

User Interface Design

The following figures show the UI snapshots for the Rescate ePCR system.

Figure 3 shows the login page that was developed for the prototype. The authentication of users was implemented; credentials are checked against the backend database. The user can be directed to the registration page upon clicking "Register".
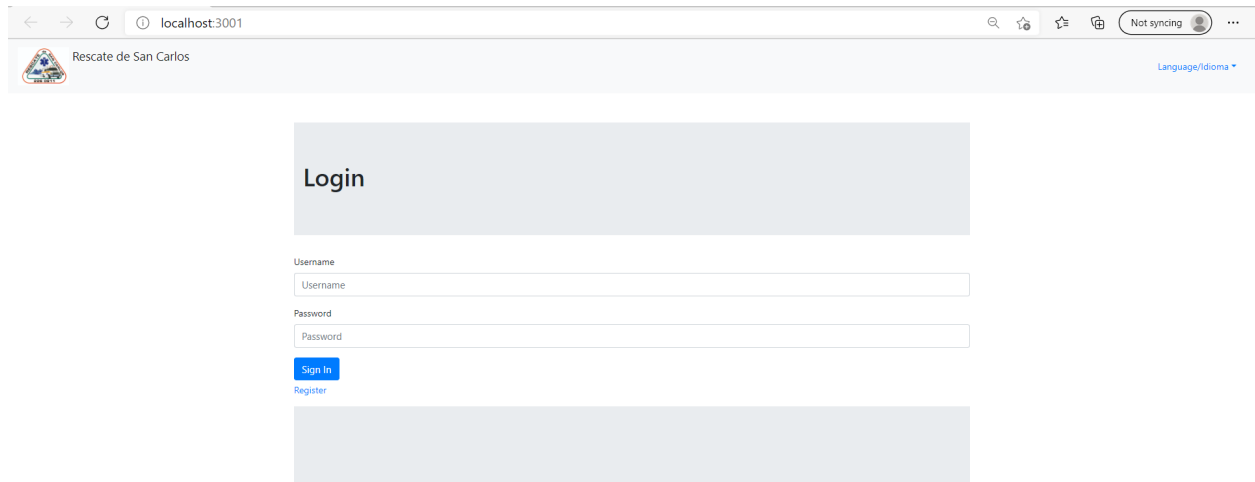
Figure 3: The login page for the Rescate ePCR system. The frontend of the page is synchronized with the backend to allow authentication.

Figure 4 shows the dropdown menu on the login/registration page to toggle the language between English and Spanish.
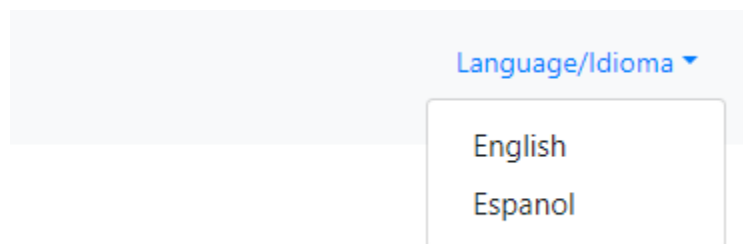
Figure 4: The menu on the login page to toggle the language of the system.

Figure 5 shows the login page after the language has been toggled to Spanish.

Acceso

Nombre de usuario

Nombre de usuario

Contraseña

Contraseña

Iniciar sesión

Registrarse
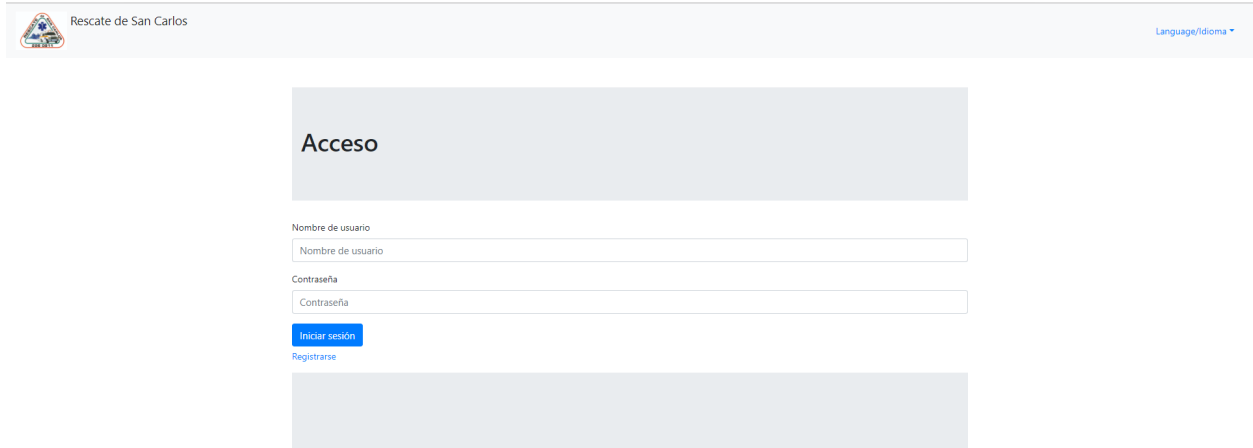
Figure 5: The login page rendered in Spanish with the language toggled.

Figure 6 shows the login message that is displayed when credentials that are not in the database are entered.

Sign In

Register

Invalid Username or password
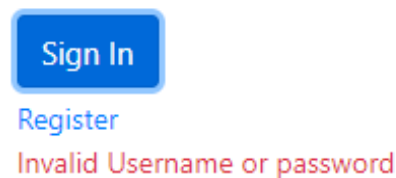
Figure 6: The message displayed on the login page when invalid credentials are entered.

Figure 7 shows the Spanish login message that is displayed when credentials that are not in the database are entered.

Iniciar sesión

Registrarse

Usuario o contraseña invalido
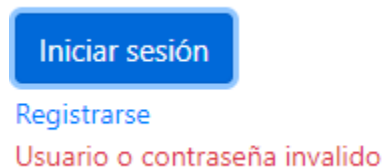
Figure 7: The message that is displayed for invalid credentials if the user is in Spanish mode.

Figure 8 displays an error message on the login page when all of the mandatory fields are not inputted.

Sign In

Register

Enter all fields
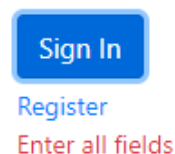
Figure 8: Blanks in the username or password fields.

Figure 9 displays an error message on the login page in Spanish mode when all of the mandatory fields are not entered.
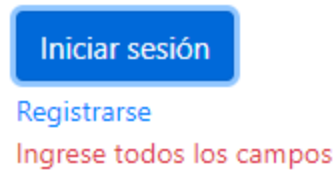
Figure 9: The Spanish version of a submission with blanks in the username or password fields.

Figure 10 shows the registration page; the login page can be cycled back to from the registration page. The registration page only allows users to register if their email is stored in the database. The admin user will be responsible for adding emails of users that can register with the system

Figure 10: The registration page for the Rescate ePCR system. The frontend of the page is synchronized with the backend to only allow valid users to register.

Figure 11 displays an error message on the registration page when all of the mandatory fields are not inputted.
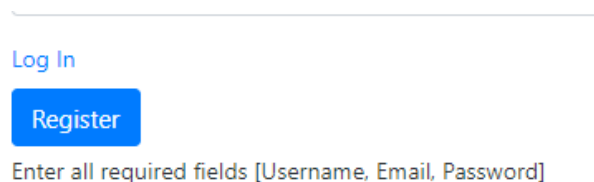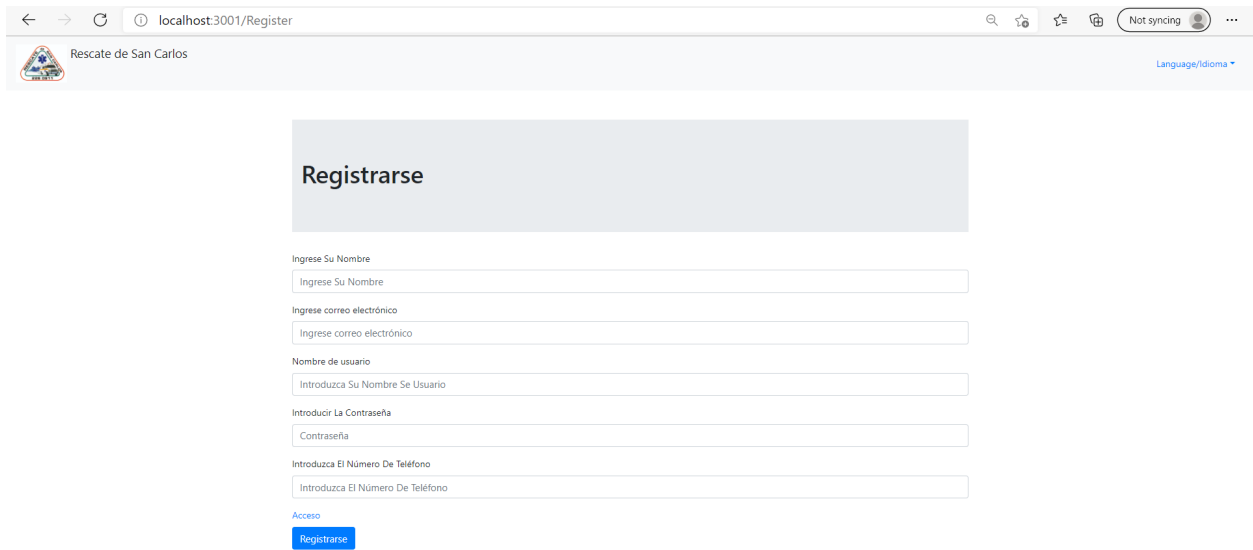
Figure 11: The error message displayed when a user does not enter all of the required fields on the registration page.
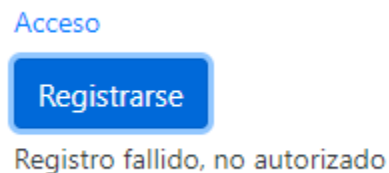
Figure 12 shows the registration page rendered in Spanish if the user is in Spanish mode.

Figure 12: The registration page rendered in Spanish. The language was toggled to render the page.

Figure 13 displays an error message on the registration page when a user that is not validated for account creation tries to register. The language is toggled to Spanish mode.

Figure 13: The error message displayed when a user does not enter all of the required fields on the registration page.

Figure 14 shows the settings page for a user that is an admin rendered in English. All users have the ability to change their account details. Admin users have the added functionality of being able to add, delete, and elevate users.

Figure 14: The settings page for a user that is an admin. Standard users do not have access to admin settings.

Figure 15 shows the button that is on the bottom of the settings page for a user to delete their own account. When a user deletes their account, they lose their permissions and are redirected to the login page.

Figure 15: The button a standard or admin user uses to delete their account.

Figure 16 shows the current navigation bar on the bottom of the webpage in the "charts" page.

Figure 16: The chart navigation bar to navigate to different sections of the form.

Figure 17 shows the updated call information section of the form with all the required details and inputs regarding the incident.

Figure 17: The call information section of the chart form split into details and times headings. Depending on the inputs typed and selected, certain parts of the field will render. For example, the MCI and Vehicle Accident checkbox will prompt the user to fill in more fields if those are checked.

Figure 18 shows the updated patient information section of the form.

Figure 18: At the top of the patient information section of the chart form, a button asks for the user to search for a previous existing patient on the database. The function is currently not implemented yet, but when the patient exists, the data from the table for that specific patient will be autofilled in some of the inputs shown on the form.

Figure 19 shows the interventions and treatment section of the form.

INTERVENTIONS / TREATMENT

PROCEDURES

Add

Procedure

MEDICATIONS

Add

Previous   Next

Figure 19: The interventions and treatment section of the chart form has not gone through some updates yet. The user is able to type a procedure name on the only input field found here. The "add" buttons will pop up a window for the user to add specific procedures and or medications in the future.

The popup windows shown in Figure 20 and Figure 21 portray the contents of the subforms used to push in data about the interventions or treatment provided to patients into the main chart form. Figure 20 is a template for adding procedures performed on the user. Depending on the procedure chosen, different input fields may be asked. As for Figure 21, the adding medications form consists of most of the input fields needed for implementation. The two figures are implemented as a design as of now without any functionality.

ADD PROCEDURE

Procedure
TBA
TBA
TBA
TBA
TBA

Time   mm/dd/yyyy --:-- --

Crew   John Doe

Note: Crew member name should be selected from current user's information. Currently, the below buttons do nothing yet.

Add   Cancel

Figure 20: The popup window for adding a procedure form. From the list of procedures, clicking on a specific procedure will render different input fields associated with that procedure. The crew member's name is automatic and retrieved from the currently logged in user.

Figure 21: The popup window for adding a medication form. The crew member's name is automatic and retrieved from the currently logged in user.

Figure 22 shows the updated confirm chart report page, the last section of the forms.

Figure 22: The confirm chart report section with all the current details from the previous sections saved. They are organized in a table that would be simple to look at.

Glossary of Terms

Admin - Short for administrator, a person who has control over a system.

Android Application - A software designed to run on Android devices or Android emulator. This software will take the form of an app and will be accessible from one of the app stores offered on Android devices.

Authentication - The process of verifying and assuring the identity of the party as the claimed one. A common method of this use is allowing users to enter a username and password.

Authorize - The process of giving privileges to access an application.

EMT - Emergency medical technician, the most common type of providers in the emergency medical services field. EMTs are often first responders for emergency medical assistance. They assess a patient's condition and determine a course of treatment.

Encryption - A method of converting sensitive data into a coded format to prevent unauthorized sources from viewing.

ePCR - Electronic Patient Care Reporting, a system that allows EMTs and healthcare providers to electronically fill out a patient chart.

Interventions - An essential element found in patient reports that shows what had been performed and given to the patient.

Online security - The process of protecting data or information that is stored on the Internet. This is necessary to protect a user's financial, medical, or personal information.

Medication - A type of treatment, such as medicine or drugs, given to patients.

Narrative - A description of the event for which the EMT was called, and what was done by the EMT (if applicable).

Notes - An addendum to a chart added by an EMT or system user. Notes are time stamped and author marked. They may not be edited later.

Patient - A person who is receiving or registering to receive medical treatment from a user of the system. Patients in this system will most often be treated by EMTs using the system.

Patient chart - A record of a patient's key clinical data and medical history, such as demographics, vital signs, diagnoses, medications, treatment, notes, problems, allergies, and lab and test results.

PDF - Portable document format, a way of displaying pages digitally which makes printing to a physical medium easier.

Procedure - A type of action that is performed on a patient as a means to achieve results that include improving health or making diagnoses.

User - A user is any person that can log on to the deployed system. Most users will be EMTs but some users could be medical staff who work around EMTs such as nurses and doctors.

User interface - The visual area to interact with an application.

Vital signs - Important medical signs that medical professionals and health care providers use to monitor a patient. These signs include, but are not limited to: body temperature, pulse rate, respiration rate, and blood pressure

Web application - A website which provides application access through a web browser, as opposed to an application installed on a system.

Engineering Standards and Technologies

MySQL - A database management tool that conforms to the ANSI SQL standard. The team will use mysql to create and maintain data in databases in various different tables throughout the project.

i18-next - A framework for internationalization that is cross platform and supports multiple languages and technologies. The team will use i18-next to implement the multilingual functionality.

Javascript ES6 - A programming language used heavily in web applications. The team will use the NodeJS runtime environment for javascript on the backend and the ReactJS javascript framework to design the frontend.

ISO 8601 DateTime Standard - An international date time and date interchange standard. This standard specifies how dates are represented and stored and how dates and times should be translated. This will be useful to the team since the project will need to support international dates.

W3C Mobile Graphics Web Application Standard - This standard defines best practices for designing mobile facing web applications. It specifies technologies to use and techniques to maintain accessibility on small screens. The team plans on referencing this standard to keep the systems web applications mobile friendly.

ISO/IEC 12207 - A standard that defines all the processes required for developing and maintaining high quality software systems. The leading principle of this standard is processes not stages. The team plans on using some of the processes in the Technical Management Processes and Technical Processes categories, specifically the processes defined for validation, verification, and quality assurance.

Project Impact and Context Considerations

An open source ePCR system has the potential to make a big impact on the quality of healthcare available in lower income and rural communities. Current healthcare record systems are expensive and are licensed for use. This ePCR system would be free and open source, allowing anyone around the world to use it. The system is currently being designed for specific use by the San Carlos Rescue ambulance service in Sonora Mexico. Having access to electronic charts means that their records are safe from environmental destruction such as in a fire or due to extreme weather such as flooding. However, electronic records are more vulnerable to cyber security threats and one of the biggest concerns regarding this system is around security. If this system were to be deployed in the United States, several aspects of the system would need to be in compliance with federal and state regulations such as HIPAA.

The security risks aside, having access to patient charts in a digital format could dramatically increase the quality of care that the EMTs at San Carlos can provide. Our system will help modernize the EMTs approaches to healthcare at little to no cost. Electronic charts can be accessed instantly and from anywhere. This will allow the EMTs to pull up a patient's history in seconds and will be useful in diagnosing and treating a patient. Similarly, electronic records decrease the time spent searching through paper records and increase the amount of time providing care. One goal of the system is to provide EMTs a reporting feature that will aggregate data in an easy to view way. These generated graphs and statistics will help the EMTs allocate resources and be continually prepared for the future. Creating an open source system will allow a community to maintain and improve the system which is something that big SaaS epcr systems do not have. More eyes on the system will help identify security issues and make needed changes that might be too small for the main developers. The ePCR system will have little impact on the environment and will be deployed on public servers such as AWS to reduce e-waste and cost. AWS will allow the records to stay out of reach of environmental disasters and will increase the longevity of the system.

References

Problem Domain Book

Yadav, S. C., & Singh, S. K. (2009). *Introduction to client server computing*. Daryaganj: New Age International Ltd.

This book covers the details about the client-server architecture. The book goes into depth about databases, server technology, testing, and more. This is a key component of our project as we will have to deal with both. The ePCR system will have a frontend and a backend.

Reference Articles

Mathioudakis, A., Rousalova, I., Gagnat, A., Saad, N., & Hardavella, G. "How to keep good clinical records" *Breathe*, 12(4), 396-373. doi: 10.1183/20734735.018016

This article discusses how to keep good clinical records. The article identifies why keeping records is important. It also provides several charts and figures that make it easy to understand what the most important on a clinical chart is. The article notes that there are several legal reasons to keep good records and maintain a structured record keeping practice.

Hydari, M. Z., Telang, R., & Marella, W. M. (2018;2019;). Saving patient Ryan—Can advanced electronic medical records make patient care safer? *Management Science, 65*(5), 2041-2059. doi:10.1287/mnsc.2018.3042

This article discusses the potential effects of advanced electronic medical records improving the safety of patients. The article analyzes the use of advanced electronic medical records with a dataset from the Pennsylvania Patient Safety Authority (PSA). The article discusses that the use of the advanced EMRs led to a 17.5% reduction in patient safety events.

Flanagan M. E., Saleem J. J., Millitello L. G., Russ A. L., and Doebbeling B. N. "Paper- and computer-based workarounds to electronic health record use at three benchmark institutions." *Journal of the American Medical Informatics Association*, Vol. 20, Ed. 1, 59-66. 2013. doi:10.1136/amiajnl-2012-000982

This article discusses the use of workaround techniques and methods that would be beneficial for some environments as compared to electronic medical record systems. Some institutions still prefer the need to use paper-based work, which the observed workarounds developed were for efficiency, awareness, and memory reasons. When patients needed to wait, they could accurately fill out and update their own information, while a technician might have been assigned to fill out information electronically with data recorded on paper previously. This article helps to point out the potential reasons for better EMR systems that could be improved efficiently by taking what is looked for paper-based work.

Liu, C., Chung, Y., Chen, T., & Wang, S. (2012). The enhancement of security in healthcare information systems. Journal of Medical Systems, 36(3), 1673-88. doi:http://dx.doi.org.unr.idm.olc.org/10.1007/s10916-010-9628-3

This article establishes the common methods in which unwanted access is gained in healthcare systems. For external threats, it discusses Javascript injection and unauthorized downloading of files (from erroneous network design). The article also proposes three categories of medical information security; availability, confidentiality, and integrity. Availability is very important in healthcare as it hastens use cases, speeding up treatment. Integrity and confidentiality are ensured by technological mechanisms, which is a topic that includes authentication and encryption.

Useful Resources

ReactJS Documentation

https://reactjs.org/docs/getting-started.html

A resource that will be useful for us is the ReactJS documentation website. The ReactJS framework is what we will use to build the UI of the web application. The documentation offers information on how to incorporate React into a website as well as the main components of it.

Sucuri Security

https://sucuri.net/guides/website-security/

A website that will be useful for us is the Sucuri guide on web security. Security must be a top priority for our project due to the sensitive nature of the data stored. The Sucuri guide includes website attributes that make them vulnerable to attacks. We will use this website as a resource to fortify the security of the ePCR system that we develop.

NodeJS Documentation

https://nodejs.org/en/docs/

A website that will be useful for us is the NodeJA documentation website. The NodeJS runtime environment is what we will use to build the backend of the application and will serve both the website and the app. The documentation offers information on how to use NodeJS as well as detailed examples and descriptions of specific features. The documentation will allow us to understand how different features of the language work.

Javascript Style Guides

https://github.com/airbnb/javascript, https://google.github.io/styleguide/jsguide.html

Style guides for Javascript ES6 used by the technology megacorporations. Ensures good programming practices.

Contribution of Team Members

Member	Hours	Contributions
Kennedy Anukam	3	1. Specification Summary of Changes 2. Updated Use Case Diagram 3. Updated Use Case Descriptions 4. System Level Diagram 5. Updated User Interface Design
Mason Harlan	3	1. Abstract 2. Engineering Standards & Technologies 3. Project Impact and Context Considerations 4. Updated Glossary 5. Updated Resources
Yi Jiang	3	1. Updated Technical Requirements 2. Updated Requirement Traceability Matrix 3. Design Summary of Changes 4. Updated Database Tables 5. Updated User Interface Design
Alexander Moore	3	1. Recent Project Changes 2. Specification Summary of Changes 3. Updated Program Units 4. Updated Database Tables 5. Updated Resources