# fEMR: Saving Lives Through Data Merging

Ritu Bhalodia, Mahi Choudhary, Elena Fowler, Spencer Klawans, Garrett Lew

Team ZZS

Reviewer: BJ Klingenberg and Bruno DaSilva

Created on 29 October 2020

Last Updated on 11 March 2021

JIRA Backlog

Github Repository

## 2. Introduction

### a. Overview, Problem Description, Summary, or Abstract

The Fast Electronic Medical Record (fEMR) system is an open-source Electronic Medical Records system tailored for “pop-up” clinics. fEMR allows mobile medical teams to treat patients and store the patient’s medical information such as conditions and past treatments by utilizing software to accurately maintain these patient records. fEMR is deployed as a self-contained “kit” comprising hardware and software. Each hardware kit provides its own local wifi network, to provide access to the fEMR electronic medical record system via mobile devices used by medics due to lack of internet connectivity in remote areas. Since kits are isolated in these environments, their software needs to be updated to the latest version so that the data on the kit can be uploaded. This project involves enhancing the kit to upload patient data to a central repository when it is connected to the internet. Additionally, fEMR wants the capability to merge or link patient records created in different locations that appear to document the same person. A suggested solution is to create a central database in order to merge data from multiple kits onto it. Furthermore, to address the issue of overlapping patient data each patient record should contain a globally unique identifier. The main stakeholders for this project include the President Sarah Draugelis and the Chief Technical Officer Andy. Additionally, the Senior Software Engineers on fEMR are Ken Dunlap and Kevin Zurek. Other stakeholders for team fEMR are the physicians/doctors who use the system to record the medical data.

### b. Glossary or Terminology

Glossary

| Word/Abbr. | Meaning |
|---|---|
| Data Fragmentation | When there is one patient who has two different records |
| EMR | Electronic Medical Record, health records stored in a digital format |
| Encounter | when a patient is treated; better to make a new encounter instead of making a new patient |
| fEMR | Fast Electronic Medical Record - open-source Electronic Medical Record system tailored for “pop-up” clinics |
| Formulary | The set of medicines a group brings on a trip |
| Fuzzy Matching | Inexact matching; matching on similarities, ex: Andy ~ Andie |
| I-Far Tool | Tool under development of Drexel University that is used to see which modules link together and dependencies in the source code |
| Kit | Hardware and software that is distributed to physicians/admin to set-up and use in clinics |

|  |  |
|---|---|
| Screening Event | Deploying EMR and testing equipment at community locations (like a barber shop) to check people for health conditions. The data is sent to a federal source, and sent to their doctor. Started in LA as an effective way to reach people in the community and decrease medical emergencies by catching issues early. |
| Scribe | The person who takes notes about the encounter |
| SLDM | Saving Lives through Data Merging |
| Triage | Location where doctors/nurses insert patients' general and vitals information digitally. |

## c. Context or Background

When a physician has a patient's medical history, they are able to provide better care because they have the opportunity to detect patterns and do not have to start the information collection process from scratch. This however is made quite difficult when the patients femr serves often do not have uniquely identifiable attributes, and often don't even use their real names. Because it is not probable that patients in most locations in which femr operates will obtain these unique identifiers and be willing to share that personal information, we must change the current system of matching records to account for this. We are implementing a patient record matching system to suggest potential matches on a kit to help physicians provide more complete care to their patients, without violating the patient's trust and privacy. There have been ideas to use identification systems such as an eye scanner so that patients do not have to remember any information such as an ID or provide their name. However, this proposal was rejected because many patients in vulnerable populations fear that the data would be used against them. Our fuzzy matching solution will ensure patients are not pressured into giving more personal data to femr in order to receive care, allowing for vulnerable populations to acquire care.

The patient data is currently stored on several hard drives separated by kit, and patient records are not globally unique. This poses a security risk as well as limits the efficiency of kit setup. In order to support patient matches, the correct patient data needs to be loaded onto the kit before a trip. Currently, there are few situations in which patient data from multiple trips are loaded onto the same kit, but it can happen when one party gives another permission to use that patient data if they will be traveling to the same location. By not storing patient data in one central repository, the kit setup process is not as efficient or accessible to people who may need to access that data. By creating a remote central repository, we can ensure patient data is being securely stored and accessible to the different parties that need access to it to aid in efficient kit preparation in the future. Since patient records are not globally unique, we must create a globally unique identifier in order to store records in one central repository.

**d. Goals or Product and Technical Requirements**

- Product requirements in the form of user stories (functional requirements)
    - As a physician, I want to see potential patient matches when I am inputting a patient's data, so that I can add a new patient encounter if there is a match.
    - As a volunteer, I want to have patient records from all trips accessible in a central database, so that I can easily download the data from specific previous trips onto a kit.
    - As a volunteer at a clinic, I want to periodically upload the new patient records to the central repository, so that only new and modified entries are replaced.
- Technical requirements (non-functional requirements)
    - The system shall merge and upload patient records when a kit has an active internet connection.
    - The system shall assign a globally unique key to each patient.
    - The system shall have a remote central database to store patient data.
    - The system shall remain functional with no active internet connection.
    - The system shall use fuzzy matching to suggest patients with similar names.
    - The system shall provide ranked potential patient matches when a physician tries creating a new patient with similar information.

**e. Non-Goals or Out of Scope**

- The system shall not match records at a global scope.
- The system shall not merge data on kits without an active and stable internet connection.
- Future goals
    - Database Driven Language Interface
        - English
        - Spanish
        - French
    - Interface for adding new languages
    - External testing Results and referral handoff
    - Disaster Ready
        - Flexible Networking
    - Tech Debt Overhaul
        - Currently, working with the code base is quite difficult due to the lack of test cases and poor code quality. Investing time in refactoring the existing code will improve efficiency.

**f. Future Goals - Phase 3**

- Self building kit (SBK)
    - fEMR should not have to give out physical kits to organizations going on trips. Instead, they give orgs access to a software application that sets up the fEMR kit on any device, so fEMR is no longer limited by the physical fleet of kits.
- Self registration kit — orgs register with fEMR to get the SBK automatically
    - Advanced User Profiles

    - Self Enrolling Provider
        - Field Manager Approval
- Interacting with medical devices to transfer data automatically
- Export demographic data/analytics tool to a CSV
- Include a trauma sheet for situations in which there is no need for patient continuity
    - Instead of just the triage interface, there is an interface for trauma patients as well

**g. Assumptions**

**Kit**

A stable internet connection needs to be present and accessible for the user in order for the solution to work. Additionally, a remote central repository needs to be created, so that the patient records can all be stored in one place.

For the database, the id value in the patients' table is the patient id — the integer value given to patients after being treated at a clinic.

**Users**

Some patients being treated at clinics give fake information because they don't want to be identified. We must be able to suggest accurate patient matches without the use of a correct name.

**Fuzzy Matching**

For the matching system, we're assuming the attributes that are weighted higher are more likely to be attributes that identify the patients. For example, we're inferring that a patient record with the same phone number is probably the same patient because generally, people don't change their phone numbers often. Although some patients may give disposable numbers, the matching system is made for patients who want to be identified and maintain continuity. A patient record with an exact matching sex is not weighted as highly because sex is a much broader grouping.

## 3. Solutions

### a. Existing Solution & Design

Currently, femr uses kits made up of hardware and software and sends them to remote areas. Then, the doctors and physicians establish an internet connection and record their patient data on the kits. Eventually, after the trip, the data from the kit is transferred to a hard drive. Consequently, the current solution involves relying on hardware and leaving the patient data on a hard drive with no backup. The data stored in a hard drive is not in a secure location and it is not safe to be leaving personal medical information on a random drive. Additionally, the patient records all have a patient id which distinguishes a record within a kit, but there is no globally unique id.

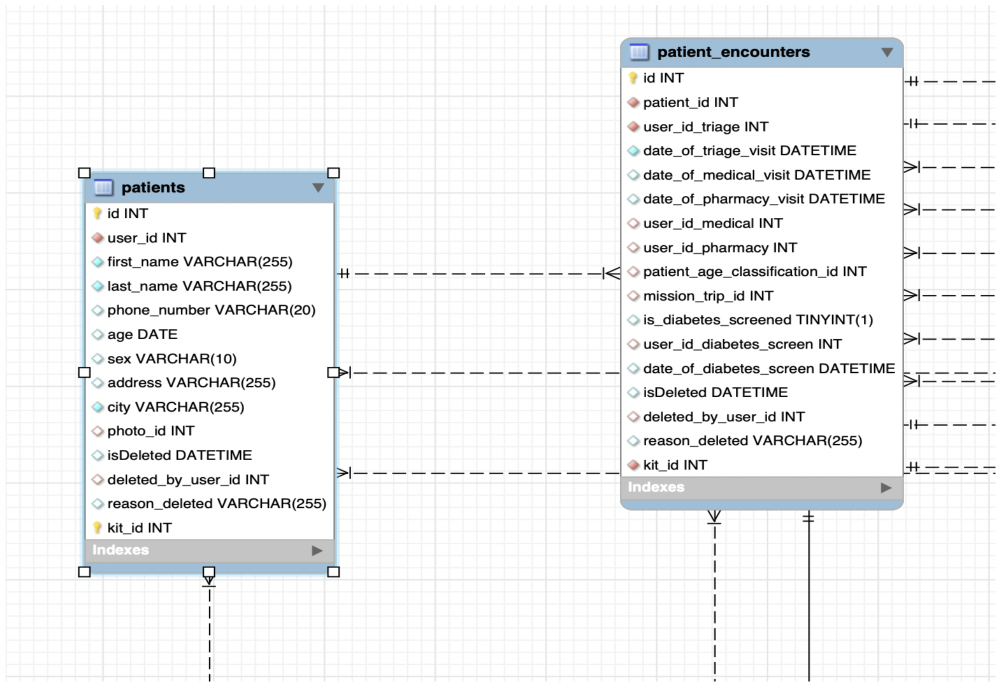
The current patient matching solution only checks for an exact match of first name and last name. This does not account for misspellings or different spellings of the same name. Additionally, it only relies on two fields from the triage form for patient matching.

**b. Suggested or Proposed Solution / Design**

- **Data Model / Schema Changes**

  We are implementing the new central repository requirement using Amazon RDS. The central repository will be updated with patient records stored in the local kits. With a central repository, it will be easier to look up a patient record because we will not have to go through the separate kits. However, since the repository holds a lot of personal data, we will need to abide by HIPPA regulations and only access the patient records with permission from the provider. The tables in this central database should be identical to those currently being used on the local kits (although may need additional administrative tables in the future).

  We are also changing the data schema of the central repo and the local kits. To uniquely identify patients (mainly for the central repo) we will be adding a column for the *kit_id* in the *patients'* table. The *kit_id* is being added to ensure the records match exactly one patient. The primary key for a patient is changed to be a compound key of (patient) *id* and the *kit_id*. Therefore any tables using the patient's *id* as a key or foreign key must be altered to reflect the changes. Currently, the only table that is related to the *patients'* table is the *patient_encounters* table. Originally *patient_encounters* uses its *patient_id* as its foreign key to relate to *patients*. Now *patient_encounters* will use a compound foreign key of *patient_id* and *kit_id* to relate to *patients*. So a *kit_id* column is added to the *patient_encounters* table.



- **Business Logic**

  There will be no API changes, but we will be utilizing the kit data to merge into the central repository. With the central repository, there may be instances where multiple patient records will contain the same information due to a patient being recorded in two different kits. There are also other logistical kit conditions that lead to an error state such as not being able to properly set up the kit and get it running. Additionally, failure scenarios will probably result from the kit not being able to securely connect to the internet, therefore the patient records will not be added to the central repository. Furthermore, only being able to build the central repository based on kit data can be seen as a limitation, as the repository is dependent on the kit to transfer the data in a safe and timely manner.

  Matching Patient Records

  We implemented a ranking system to show physicians potential patient matches when they are inputting a new patient. Below is a table indicating the point values assigned to each matching attribute. In order to be shown as a potential match, the total score must be 30 or above. The score is shown to the physician when they view the potential matches.

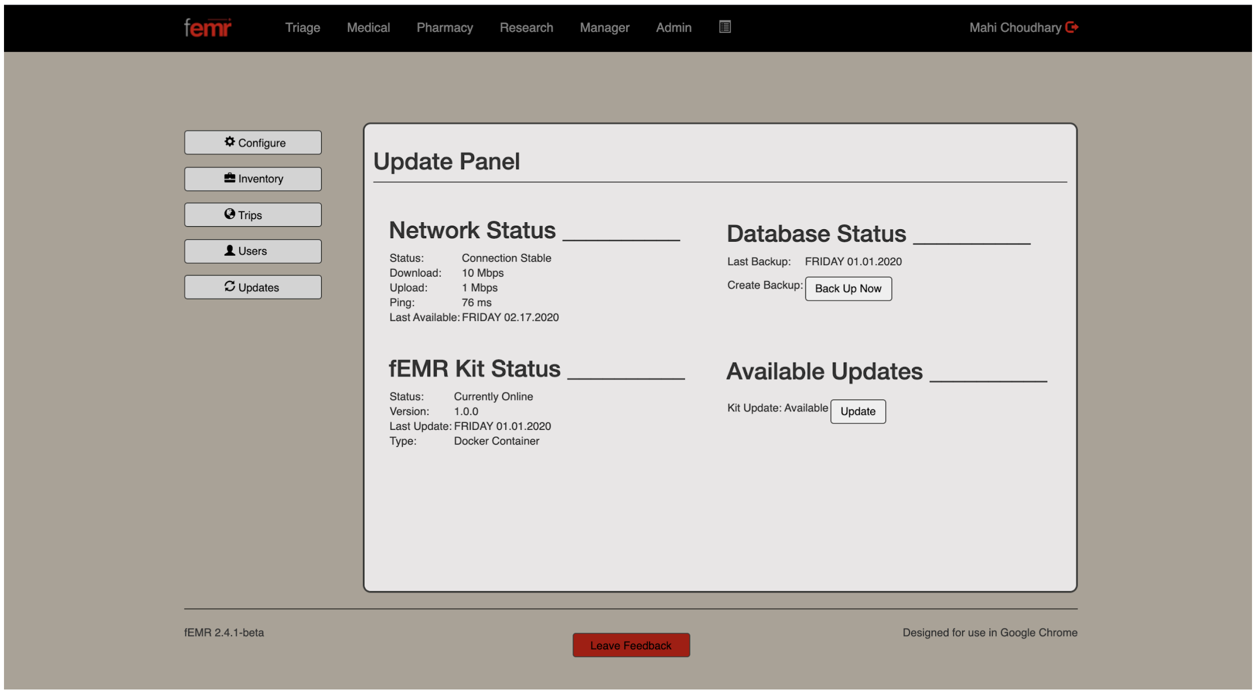
| **Attribute** | **Value** |
|---|---|
| Phone Number | 40 |
| Last Name | 15 |
| First Name | 10 |
| Last Name Double Metaphone Symbol | 10 |
| First Name Double Metaphone Symbol | 10 |
| Exact Address | 15 |
| Age | 10 |
| Gender | 10 |
| City | 10 |

  To identify misspellings and different spellings of the same first name or last name, we use the double metaphone fuzzy matching algorithm to generate a symbol for each name, then compare patients' symbols to look for matches in pronunciation. We stored the double metaphone algorithm as a function in the database using the implementation found here.

The original page shown when there appears to be duplicate patients does not allow for the patient records to be merged efficiently. Even if an existing patient is selected from the list of potential matches, a new patient record is already created using the information from the triage form. We would like to alter this behavior to only create a new patient when the provider selects it to open a new encounter.

- **Presentation Layer**

There are several improvements that were made to the UI of the application. First, we updated the background to be a solid color rather than a photo. We have changed the "leave feedback" option near the bottom of the page to be formatted exactly like the submit button on the triage page so that users can spot the button as a clear link that will lead to a page where they can leave their comments. Additionally, we have spaced out the field boxes on the General Info box which is displayed on the triage page. Then, for the Vitals Section on the triage page, we have added more checkboxes specifically "History of: diabetes, hypertension, high cholesterol, tobacco use disorder, and substance/alcohol abuse". The height of the boxes were adjusted to be equal. We also have increased the size of the navigation bar at the top of the application to improve the readability of the buttons. Lastly, on the admin page we have decided to add a section to the Admin tab named "Updates" as shown in the figure below. The updates panel will include information about the network status, fEMR kit status, database status, and available updates. Below is the newly implemented UI for the Update Panel:



Update Panel

Network Status

Status: Connection Stable
Download: 10 Mbps
Upload: 1 Mbps
Ping: 76 ms
Last Available: FRIDAY 02.17.2020

Database Status

Last Backup: FRIDAY 01.01.2020
Create Backup:

fEMR Kit Status

Status: Currently Online
Version: 1.0.0
Last Update: FRIDAY 01.01.2020
Type: Docker Container

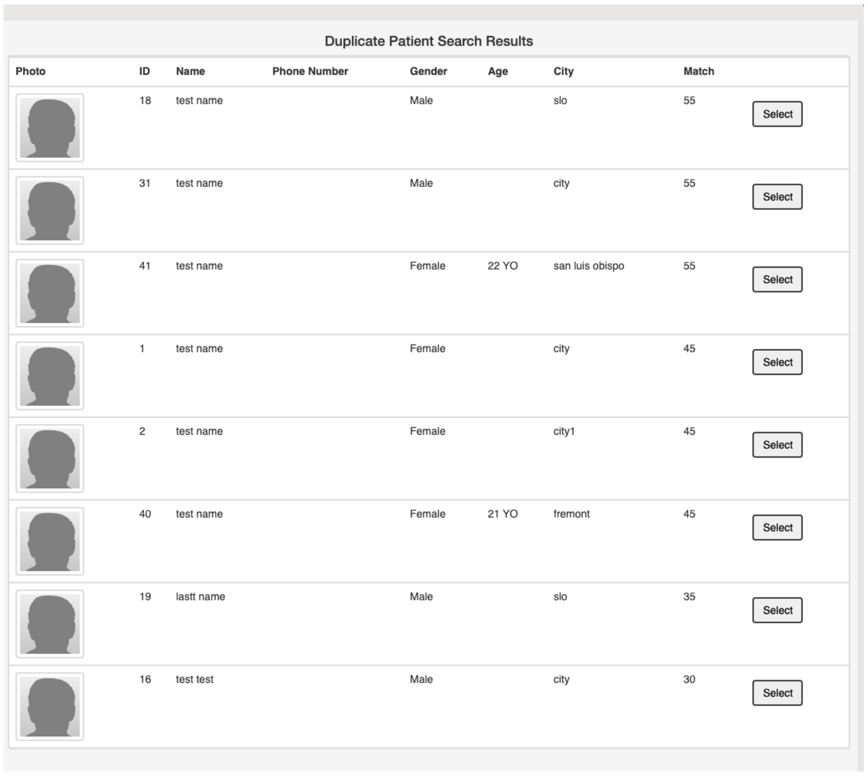
Available Updates

Kit Update: Available

fEMR 2.4.1-beta

Designed for use in Google Chrome

Concerning the patient matching, we have added a new match column on the duplicate patient search page. This match column is ordered by the highest patient matching rank score that is calculated based on the ranking query. Additionally, the page will show the top 15 matches and only display possible matches with scores above 30. Below is the UI for the implementation of rank scores associated with potential patient matches:



Duplicate Patient Search Results

| Photo | ID | Name | Phone Number | Gender | Age | City | Match | |
|---|---|---|---|---|---|---|---|---|
| | 18 | test name | | Male | | slo | 55 | Select |
| | 31 | test name | | Male | | city | 55 | Select |
| | 41 | test name | | Female | 22 YO | san luis obispo | 55 | Select |
| | 1 | test name | | Female | | city | 45 | Select |
| | 2 | test name | | Female | | city1 | 45 | Select |
| | 40 | test name | | Female | 21 YO | fremont | 45 | Select |
| | 19 | lastt name | | Male | | slo | 35 | Select |
| | 16 | test test | | Male | | city | 30 | Select |

- **Other questions to answer**
  - How will the solution scale?
    The solution is scalable because fEMR is transitioning towards a more software based kit based kit criteria compared to their current hardware business. The kit itself is built to be scalable so the admin before deploying on their trip is able to adjust the system based on their own preference/need. This requires software to be flexible and adaptable hence being also scalable for each trip's individual requirements.
    To aid in making fEMR more scalable and easier to deploy, a docker image was created so that it can be more quickly deployed. This also allows kit admins to pick which fEMR version they want if needed.
  - What are the limitations of the solution?
    The current limitations of the current solution are mentioned above as, currently there is not a way to merge data within a disconnected environment. Other limitations of getting out of the hardware business includes the changing the current norm of how users interact with fEMR, economical changes- how the nonprofit is still able to stay afloat while transitioning, and business shift for customers and their current understanding of fEMR and its uses.
  - How will it recover in the event of a failure?

In the event of failure, it seems to be that the most common failure that occurs with the kit is logistical issues. The fEMR team sets aside employee volunteers to help with these issues as they arise almost as being “on-call” to when issues happen. As for data loss on the job, this issue is not retractable as the connectivity issues also hinder the possibility of somehow saving information. The nurse practitioners and scribes sometimes retain a hardcopy of the patient information. Luckily data loss has decreased as the fEMR project has had more experience in the field. Sometimes technical bugs do come up, but the fEMR team has a system that allows users to record their issue to relay back to fEMR software folks to fix. When a transaction fails, MySQL will roll back back to the previous state before the transaction begins. This will leave the database in the state before the attempted sync/merge.

- How will it cope with future requirements?
  Coping with future requirements basically is a learn by errors method and teaching the right, good practices. If the logical issues are the most common, perhaps doing training with admin/doctors before deployment could be implemented. Future requirements are yet to come, discussions about data merging in disconnected environments seems like the ideal situation, however this is not in the scope of this project.

**c. Test Plan**

Unit tests will be implemented in order to validate that the system is running smoothly. The build for the system should complete successfully. If a build fails, then we will make sure to stop other work and prioritize it by getting the build to pass before we continue implementing other features. Currently, there are not a lot of tests, so we will make sure to add more tests for the main components such as medication service. For now, our focus will be to review the current tests and increase test coverage in general for the existing code. For the code that we contribute, the minimum bar for standard test coverage will be 90%, but it might be difficult to test some of the user interface components we will change. For the unit tests we will continue to use Junit and the mocking framework, Mockito. Additionally, we are using Github Actions to implement continuous integration in order to automate the testing.

**d. Alternate Solutions / Designs**

There are several discussions we had about our proposed solution. We considered alternative databases beyond Amazon RDS, but decided to use AWS because we get student credits and want to be frugal as we are working with a nonprofit. For the globally unique identifier for each patient record, we considered having just one field that indicates both the patient id and kit id, such as a fixed length string, instead of using a compound key. We decided against this approach because we want to be able to retrieve records by just patient id and kit id individually, and each of those values should not be limited to a fixed size or format.

We also discussed alternatives to our solution of merging patient records. We discussed focusing on merging the data at the central database level as well as at the local kit level. We decided to focus on merging data at the kit level because it prevents duplicate patient records from being created when the patient is present, so the physician will be able to access their medical history immediately. Merging data at the central repository level may be considered a future goal.

All design decisions were made with the input from fEMR representatives and in accordance with their feature priorities.

## 4. Further Considerations

### a. Security considerations

- What are the potential threats?
  Some potential security threats include that the kit deals with sensitive information and hackers would want to get this information for malicious intent. Other security consideration is that some of the patients that come in for help are scared and fearful, their trust is important to ensure they get the help they need so security needs to be reliable on each step.
- How will they be mitigated?
  Mitigation for security risks include keeping the patient information separate for each trip unless there is a patient information match that will benefit the patient in getting help based on past medical records. Also obeying the HIPAA laws and data privacy laws also helps mitigate against security. The data itself is not shared whatsoever, there is a hard-drive that is securely stored.
- How will the solution affect the security of other components, services, and systems?
  AWS RDS supports https connection, so the data will be secured while in transit between the central repo and kits. Our added solutions will not affect the security of other components (kit databases). Where the added components include the central database, and patient matching strategy.

### b. Privacy considerations

We are storing personally identifiable information and medical information, often for people in vulnerable populations. However, in most of the countries in which femr operates there are no strict privacy laws equivalent to HIPAA in the United States. To ensure we protect patient privacy and uphold community trust, we will strive to uphold HIPAA even though it is not required by law. In order to do this, although the central database will contain information from multiple kits and medical providers, providers will not be able to download and view data from a kit that they did not work on previously without explicit permission from the previous provider. Additionally, by storing the data in a central database hosted by Amazon, we are lowering the risks posed by keeping physical harddrives in the hands of an individual who is not a provider. When accessing the data, a HTTPS connection will be made, so the data will be encrypted in transport. Amazon RDS provides an option to encrypt the data at rest within the database as well. We will be utilizing this option to ensure the data is encrypted at all times, at rest and during transport.

### c. Regional considerations

Kits will still be self contained unless a provider gives explicit permission for another provider to access their patients' data. By creating a central repository, downloading data onto a kit is made easier because there is one central location from which to gather the data, but the privacy of patients remains with the provider because the kit id, and therefore provider information, is attached to every patient record. Patient data will not be made available to more people through the central database, so there are no changes to the distribution of information across regions. The system currently operates in only one language, despite kits being sent around the world. Creating plans for localization with languages and unit systems is out of scope for the current project.

**d. Accessibility considerations**

- How accessible is the solution?
  Currently, there are clear accessibility violations according to the Web Content Accessibility Guidelines (WCAG). For example, some text and background colors, specifically red over black, are low contrast which may make it difficult for some people to use the product. From our interactions with the team and product, there does not appear to have been any practices in place to ensure compliance with WCAG or ADA guidelines as different students and developers were contributing to this project. In addition to implementing our backend solutions, we are committed to improving the existing UI to create a more accessible user experience.
- What tools will you use to evaluate its accessibility?
  We will use both the American Disabilities Act (ADA) guidelines and the Web Content Accessibility Guidelines (WCAG) to evaluate our product's accessibility. Descriptions of the WCAG standards can be found here.

**e. Operational considerations**

- Does this solution cause adverse aftereffects?
  - If connection drops, data can't be synced to the central database, however the local database is maintained.
- How will data be recovered in case of failure?
  - The data will not be recovered in case of failure to sync to the database.
  - If the physical kit is destroyed, data will not be recovered.
  - If connection is lost, the upload won't complete and the upload process will standby until connection is restored.
- How will the solution recover in case of a failure?
  - Manual recovery of data if possible.
- How will operational costs be kept low while delivering increased value to the users?
  - Only syncing data when needed to reduce connections to the database, therefore reducing costs.

**f. Risks**

- What risks are being undertaken with this solution?

Some risks include not being able to match patient information guaranteed during a different deployment of the kit, or if the central database somehow loses data. Another possible risk is losing the physical hard-drive that contains all the back-up information.

- Are there risks that once taken can’t be walked back?
  If the risk of data loss occurs and we can't match patient records this may not be able to be walked back because the patient’s life may be in danger.
- What is the cost-benefit analysis of taking these risks?
  The costs of using the AWS database can become relatively expensive through the use of constantly updating the data however it is hard to justify whether it is economically worth the risk due to the patient/refugee’s life being at stake. The ultimate goal is to save lives, but as a non-profit there is a need to make sure money is spent accordingly.

## 5. Deliberation

### a. Discussion

All members of the team agree with the solution.

### a1. Team Practices

As a team we use Agile practices. For most tickets, we use a lead and co system to incorporate multiple people (a lead is assigned to close the ticket, and co(s) are assigned for assistance in working on the task). This allows for more collaboration and for keeping each other accountable. For large design changes, we use a committee system. Groups of two or three people will meet up outside of the main group to deliberate or research the designs or changes so that we produce solutions that the team agrees on. Then committees will present their findings and decisions to the main group where further discussion can take place. We also hold stand-up meetings every Monday, and sometimes on Wednesday to keep each other updated on the progress throughout the sprint and to discuss different solutions.

### b. Open Questions

- How should we resolve the issue of when a patient has two different existing records?

- After a trip, when/how will the kit data be uploaded to the central repository?

### 8. References and Acks

Sarah Draugelis, Ken Dunlap, Andy Mastie and Bruno da Silva have been very helpful answering our questions about aspects of the design and how it may be implemented.