Self-Enrollment and Self-Building Kit
Tech Spec

Henry Pigg, Walden Hillegass, Conner Reavill,
Ryan Frank-Kalin, and Matthew Hung

Created on: Nov 4, 2022
Last updated: Nov 4, 2022

Jira: https://platinum.cscaws.com:8443/projects/FIB/summary
Github: https://github.com/orgs/CPSECapstone/teams/fibula

Table of Contents

Introduction

Summary

As fEMR is adopted by an increasingly larger user-base, the ability for the platform to enroll and verify new users should be as frictionless as possible. While the system at present works well enough, as the scope of fEMR increases the individuals in charge of manually entering the information for new enrollments, as well as sending hardware to prospective clients should have some of the burden removed from their shoulders. To this end our team is

creating a system that will allow individuals to enter their information into an enrollment system, and download everything they will need to utilize fEMR's software in the field.

Glossary

- ➔ **fEMR** - Fast Electronic Medical Records; the organization that we are helping.
- ➔ **Kit** - the fEMR software
- ➔ **Self-enrollment** - the process where clients can fill out a form to request a kit
- ➔ **ECR** - Elastic Container Registry
- ➔ **HIPAA** - Health Insurance Portability and Accountability Act
- ➔ **On-Chain** - On-Chain is fEMR's online EMR software
- ➔ **Off-Chain** - Off-Chain refers to the offline EMR kit
- ➔ **SAAS -** Software As A Service

Background

The problem we are solving is a problem that every client of fEMR off-chain experiences. fEMR is a software company, however in order to provide the software to the providers, they have needed to act as a hardware rental company. This makes things difficult for both fEMR and the client. The client has to wait weeks before receiving the off-chain software, and fEMR needs to maintain an inventory of computers and handle shipping and returns. With our solution, the client will be able to download the off-chain software directly to their computer and cut out the hardware from fEMR's business.

Product and Technical Requirements

Product Requirements

- As a fEMR administrator, I want to see enough information to verify self-enrolled clients and grant them access to the system.
- As a non-profit health organization, I want to enroll my health workers onto fEMR.
- As an Aid Worker preparing to deploy to the field, I want to be able to set up a fEMR system myself.

Technical Requirements

- The system shall be operating system agnostic.
- The system shall be hosted in Docker and installable by non technical people.
- The system installer shall allow users to sign in with a username and password.
    - The reason being, is we want to ensure that unauthorized users cannot install the EMR technologies without being verified by FEMR

Non-Goals or Out of Scope

- Payment for the software

Future Goals

- Internationalization of software (translate language, metrics, etc.)

Assumptions/Dependencies

- All medical providers have access to a laptop or device that is able to run and installer Docker
- Providers are able to connect to the internet for around 15 minutes in order to install the software
- Our Minimum laptop: TBD
    - We want to ensure the requirements and limitations of each installer and the kit itself before we officially settle on a minimum viable laptop. Currently we are planning on using the newer levels of OS for all operating systems.
- fEMR needs an Apple developer license in order to make an installer.
- Providers are able to install Docker previous to running the installer by following our documentation steps.

Solutions

Current Solution

Currently, fEMR has an offline solution, but it involves a tedious approval process and requires the client to request hardware from fEMR with the software already installed. The client must first email Sarah Draugelis, the CEO of fEMR to request the software, then Sarah has to manually approve and create the appropriate campaign in fEMR. Then, the hardware containing the kit and other necessary software is shipped to the medical providers which they can then take out into the field. Once they are able to connect back to the internet, all the data is then uploaded to the database.

This current solution works, however its main drawback is speed and time. The time it takes to ship the hardware can be a major limiting factor and can prevent the medical providers from providing urgent aid. There is also a lot of work on the fEMR side as well - the approval, creation of the campaign, shipping, etc. In addition, fEMR needs to maintain an inventory of computers and handle shipping and returns. All of this can be cut down in order to have minimal amounts of work for the admins at fEMR.

Proposed Solution

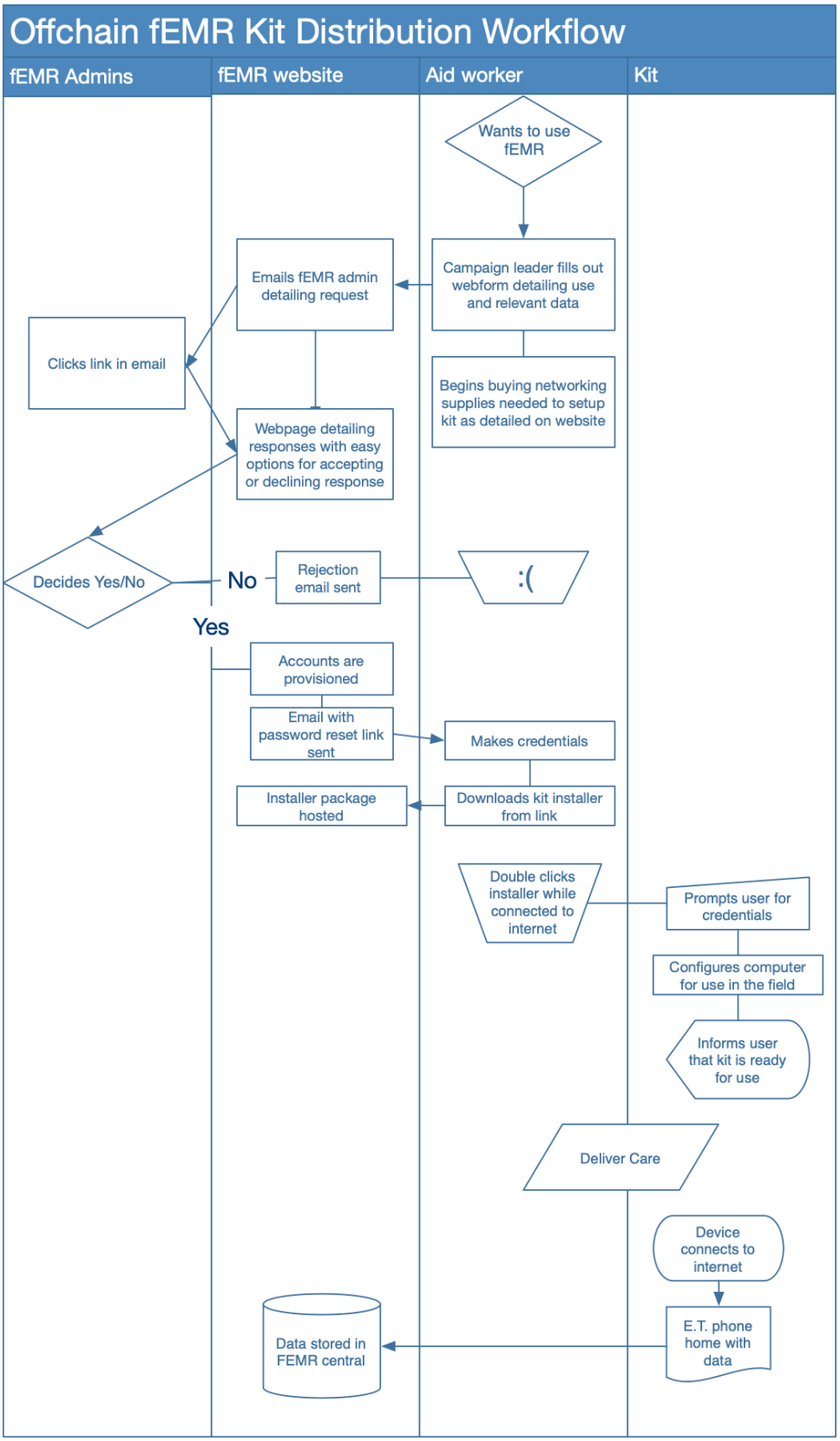
Solution

Our proposed solution is a self-enrollment page within the fEMR website that allows prospective clients to sign up for fEMR Off-Chain. When a client signs up, fEMR admins receive an email with the client's access request and can either approve or deny the request. Once approved, the client will be prompted to create an account and given the option to download the fEMR Off-Chain installer. This installer will set up the development kit and allow the client to use fEMR's software in locations without internet access.

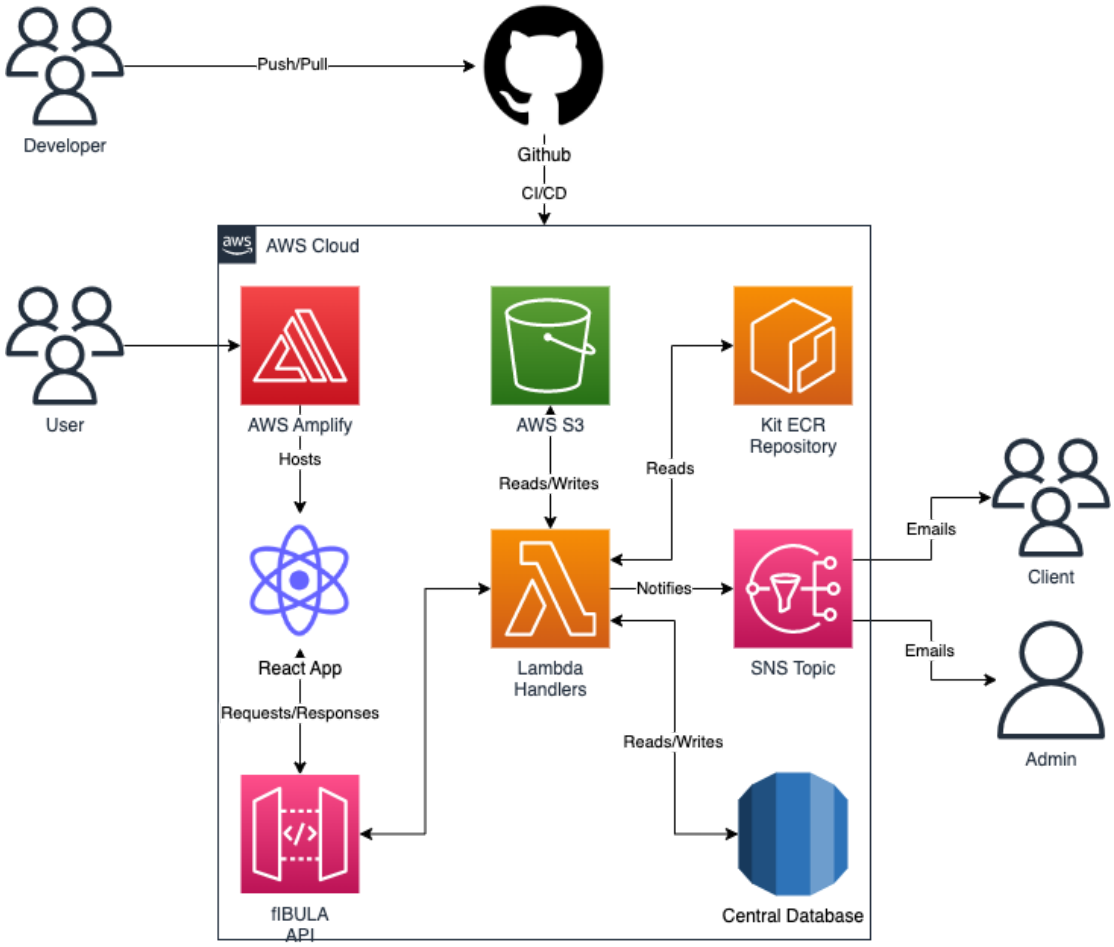
Tech Stack

- AWS Services
    - Amplify
        - We will use Amplify to deploy our React app.
    - S3 (Simple Storage Service)
        - We will store installers in S3 for download.
    - API Gateway
        - We will use API Gateway as our backend API.
    - Lambda
        - The endpoints in our API will be backed with Lambda handlers.
    - ECR (Elastic Container Registry)
        - The Off-Chain docker containers will be stored in ECR
    - SNS (Simple Notification Service)
        - We will use SNS to send emails to clients and admins for self-enrollment.
- Other External Services
    - InstallAnywhere
        - We plan to use InstallAnywhere to create an installer for the Off-Chain kits.
    - GitHub Actions
        - We will use GitHub actions along with some built-in tools in AWS for CI/CD

Use Case Workflow Diagram



Architecture Diagram



Test Plan

We will be using various forms of testing on different parts of our codebase. We are using unit tests on everything we can, and when relevant, we will mock functions and databases to ensure consistency. We are also using an AWS pipeline to release our installer. The backend will mock the database to ensure HIPAA laws and to ensure consistency. For the installer itself, we will be doing manual tests on the package on multiple different operating systems.

Alternate Solutions

An alternate solution if the issue of supporting a wide range of platforms would be to decouple the installer from the self enrollment kit, and instead provide instructions on how to set up the kit. This would allow users to attempt to install the kit after enrolling, though that would increase the risk of error and prospective clients being unable to install the software successfully.

Further Considerations

Security Considerations

fEMR relies on small revenues from the licensing of the fEMR kits to aid providers. They would like to continue receiving these revenues. As a result, fEMR would like to approve everyone before they are provisioned an account, and require that this account be authenticated before the installer runs.

We should be confident that the installer we are running in the process of provisioning kits is free of malware. We will do this by enforcing authentication to push to the ECR and updating the installer.

The installation process will randomize the database and superuser password when being run.

Privacy Considerations

Patient medical records are very sensitive data. The United States has strong HIPAA laws that make developers and companies criminally liable for exposing patient information. We must work judiciously to keep this data safe and ensure that read access to the central database with non-anonymized patient information is strictly controlled by developers with HIPAA training.

Regional Considerations

Currently, the entire fEMR system is only available in English. As a result, we do not think it is necessary to implement other languages for the installer. However, there have been conversations about adding multiple language support to the fEMR app. If we move forward with this, the installer and web interfaces we use will need to be regionalized.

Accessibility Considerations

The self enrollment kit will be designed to be used by individuals with basic understanding of computers, with no technical skills required. The installation of software will also be handled entirely by the program itself, with no need to manually install Docker or any of the other components of the kit. We will also engage in user testing to evaluate the ease of use of the software as well.

Operational Considerations

Since the system is designed to function in a region with no internet, it is essential that our installer reliably works and creates a usable instance of the fEMR system. We will work with the expectation that users will have internet access during installation, and after care has been delivered, but otherwise will be fully offline for the duration of the mission. There needs to be clear user feedback if the install fails and the installer should clean up after itself so that the install can be retried safely.

The process we prescribe to care providers needs to be simple enough so that they can set up the system without any sort of specialized IT training.

Deliberation

Discussion

Our previous discussion around installer packages has been resolved and we have chosen ones to use for each platform. On the Windows side of things, we are using AdvancedInstaller due to the free non-profit organizational license. For MacOS, we are using Apple's installer.

Open Questions

Previously, we had many open questions regarding installers that have now been resolved or decided on. One of these decisions we made was regarding operations systems in the browser and determining how to provide them the correct installer download link. The page will contain multiple links separated by OS and the user will be able to choose the correct one accordingly. As mentioned in the above section, the question of budgeting for an installer software has been resolved due to the free non-profit organizational license.

Some questions from before that we haven't had the opportunity to dive in are regarding installer testing and user user permissions. We are still discussing better ways to test the installer besides having to do it manually, but we have at least set up a continuous integration setup. We are concerned about how user permissions will be handled and which team will create and maintain the user architecture.

References and Acknowledgements

References

Acknowledgments

- BJ Klingenberg - California Polytechnic State University CSC/SE Department
- Sarah Draugelis - President & Co-founder at Team fEMR
- Andrew Mastie - Chief Technology Officer at FEMR