MULTI-SITE EMR SURVEILLANCE AND CLINICAL DECISION SUPPORT FOR HIV **PREEXPOSURE PROPHYLAXIS (PrEP)**

Presented By IT Crowd Team
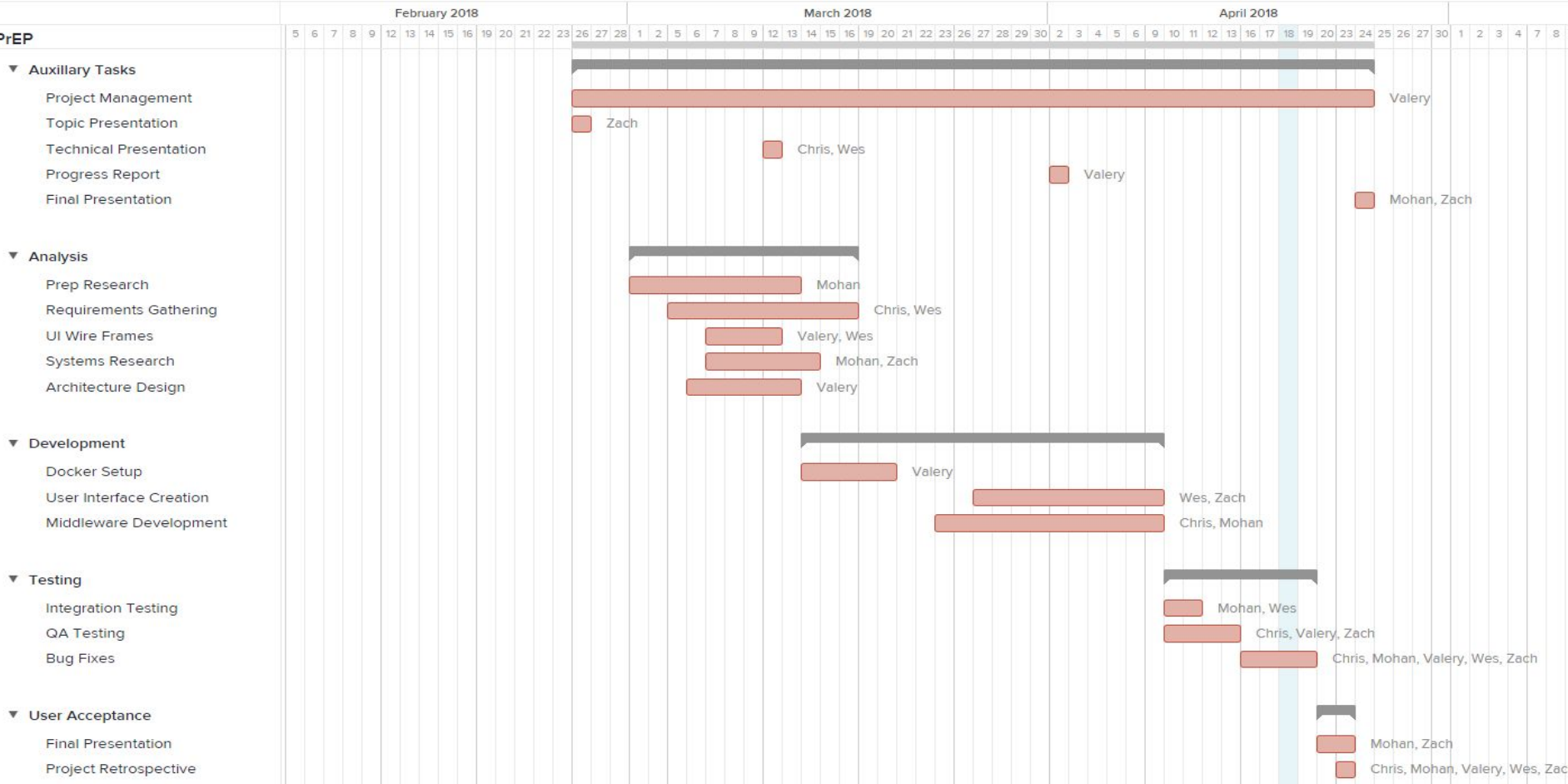
April 21, 2018

- Chris Laboy - QA
- Mohan Gangadhar Gudey - Developer
- Valery Samovich - PM
- Wesley Tomjack - Developer
- Zach Hooks - Developer

Streaming source

YouTube link to presentation:

- https://youtu.be/OEce_kkytDM

Gantt Chart

PrEP	February 2018	March 2018	April 2018

Auxillary Tasks
- Project Management — Valery
- Topic Presentation — Zach
- Technical Presentation — Chris, Wes
- Progress Report — Valery
- Final Presentation — Mohan, Zach

Analysis
- Prep Research — Mohan
- Requirements Gathering — Chris, Wes
- UI Wire Frames — Valery, Wes
- Systems Research — Mohan, Zach
- Architecture Design — Valery

Development
- Docker Setup — Valery
- User Interface Creation — Wes, Zach
- Middleware Development — Chris, Mohan

Testing
- Integration Testing — Mohan, Wes
- QA Testing — Chris, Valery, Zach
- Bug Fixes — Chris, Mohan, Valery, Wes, Zach

User Acceptance
- Final Presentation — Mohan, Zach
- Project Retrospective — Chris, Mohan, Valery, Wes, Zach

Progress Status

Project Status

- UI is completed. Both pages for Surveillance and PrEP recommendations are developed.
- Web service is also developed based on requirements. The logic for both algorithms has been implemented based on available information and standards.
- Testing complete.
- Outstanding elements (Mentor engagement and FHIR standard limitations)
  - HIV partner and MSM (Algorithm A)
    - No FHIR Resource
 - nPEP and ARV (Algorithm B)
    - Lacked mentor clarification

Research

- Medical
  - Research ICD-10 codes for Conditions such as Chlamydia, Gonorrhea, and Syphilis.
  - Research LOINC codes for Diagnostic Report tests such as Chlamydia, Gonorrhea, Syphilis, Hepatitis B, and HIV.
    - Data location in git repo in .csv: /service/src/main/resources
  - Research for HIV
- Technical
  - Research how implement CORS.
  - Research on Docker (Dockerfile, docker-compose).

Functionality

App fulfill the design proposed by the team and fully meet user requirements.

- **Packaging**: Linux Based Docker container.
  - Dockerfile
  - docker-compose
- **UI**:
  - HTML
  - CSS, Bootstrap
  - Javascript, Ajax, jQuery
- **MIddleware**:
  - Java,
  - Spring Boot,
  - REST API.
- **FHIR Server**

Note: **Lucidchart** was used to create a component diagram.
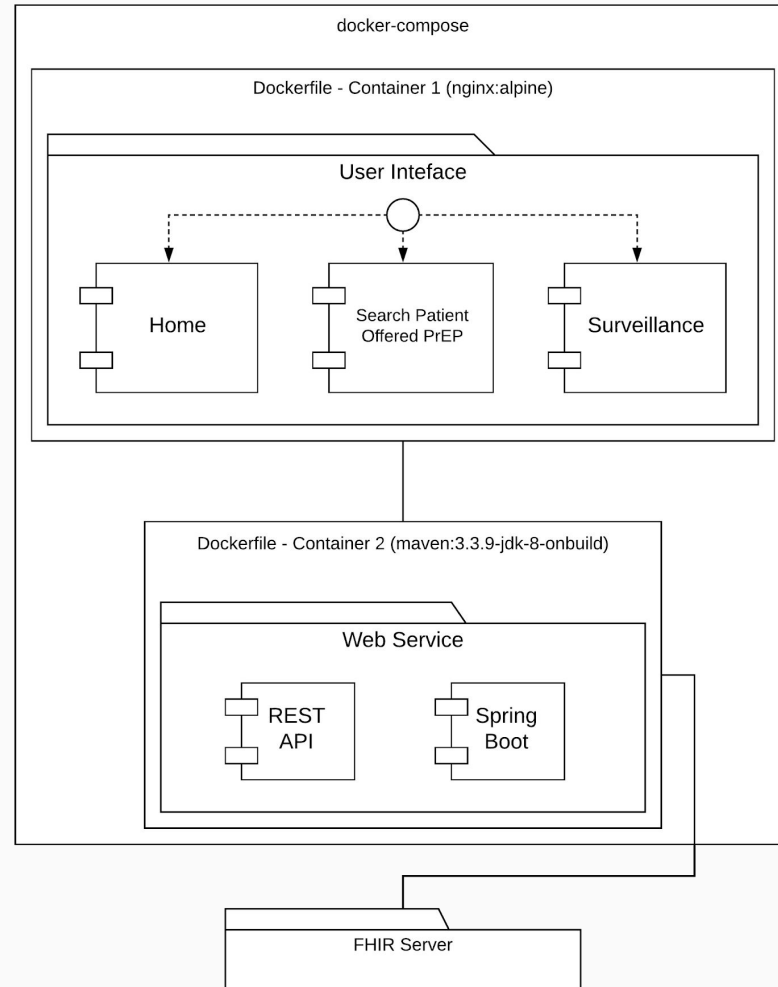
Usability

- Straight forward user interface
- Technical documentation also was design as part of the project. Its contain 3 section:
  - Installation
  - Usage
  - Contribution
- User Manual documentation also include 3 sections:
  - Navigation
  - Search Functionality
  - Surveillance

IT Crowd Team

MULTI-SITE-EMR-SURVEILLANCE

MULTI-SITE EMR SURVEILLANCE AND CLINICAL DECISION SUPPORT FOR HIV PREEXPOSURE PROPHYLAXIS (PrEP).

Installation

- Prerequisites
  - Docker
  - Maven
  - JDK-1.8

Usage

To run application from **Command-Line Interface** (CLI) via `docker-compose`

- Build the application through your CLI: `docker-compose up`
- Use `rebuild.sh` to stop, rebuild and start containers.
- WARNING: Use `clean.sh` to kill and remove containers and images.

To run **Website** separately via **Dockerfile**

- `cd` in website directory
- To build image for website run: `docker build -t <image-name> .`
  - Example: `docker build -t website .`
- To run docker image on container: `docker run -p 80:80 <image-id>`
  - `-p` port argument can be specified any
  - `-d` argument can be used with detached mode
  - Example: `docker run -p 80:80 website`
- After successful build & deploy, website can be accessed on `http://`

To start **PrEP Web service** separately via **Dockerfile**

- `cd` in service folder
- To build image for website run: `docker build -t <image-name> .`
  - Example: `docker build -t service .`
- To run docker image on container: `docker run -p 8080:8080 <image-`
  - `-p` port argument can be specified any
  - `-d` argument can be used with detached mode
  - Example: `docker run -p 8080:8080 service`
- After successful build & deploy, website can be accessed on `http://localhost:8080/<endpoint>`
- Example endpoints:
  - http://localhost:8080/getPatientDetailsList?fromDate="2018-08-16"&toDate="2018-08-16"
  - http://localhost:8080/getPatientById?id=1116172

To validate successful build with maven: `mvn clean verify`

Contribution

RECOMMENDED WAY. NOT REQUIRED: Please contribute using Github Flow. Create a fork, add commits, and open a pull request.

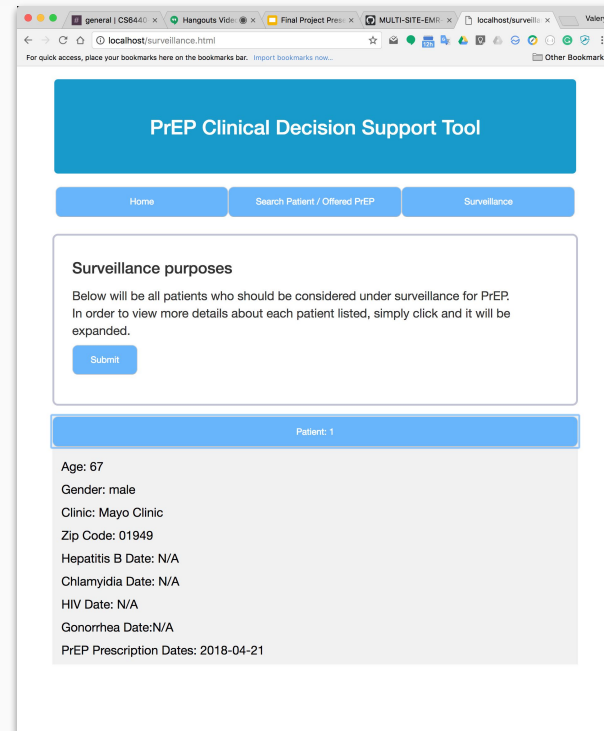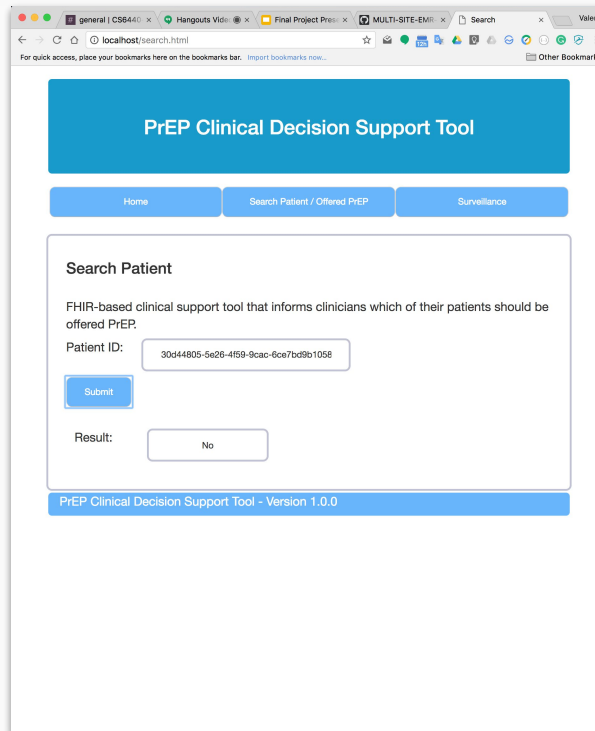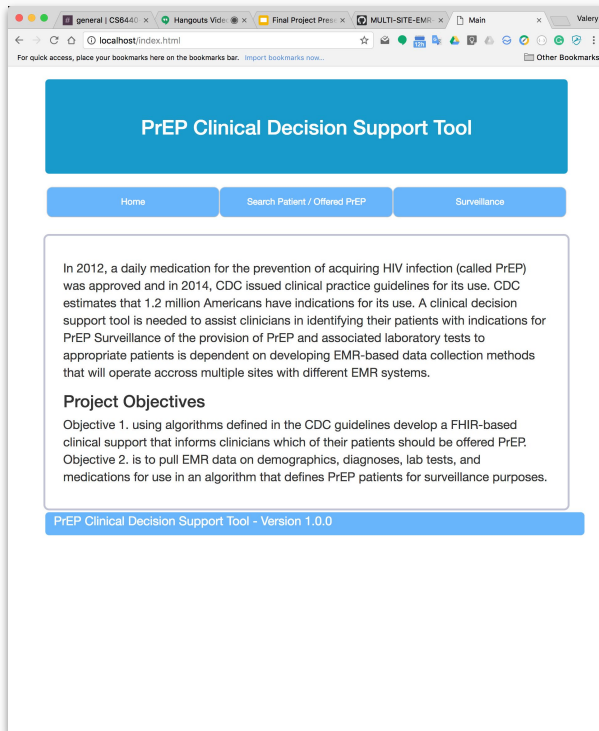## User Manual

1. **Navigation**
   i. Once logged in a user will have the ability to visit 3 different pages as part of our application (Home, Search, Surveillance).
   ii. In order to navigate to different pages, there are 3 buttons under the header on every page. Users will use those in order to toggle between each page.
2. **Search Functionality**
   i. When a user is on the search page, they will have the ability to find out whether or patient should be recommended for PrEP.
   ii. User will insert the patient id of the patient in which they would like the applications determination.
   iii. There is a result field at the bottom that will display the recommendation of Yes or No, or in the cases where a patient isn't found "No Patient".
3. **Surveillance Functionality**
   i. When a user is on the surveillance page they will have the ability to find all of the patients who should be under surveillance.
   ii. In order to do so User's will click submit.
   iii. Returned below will be a list of patients based on a 30 day window, and in order to view details of each patient the user will need to click on the patient and it will open up.

Design

Resources

Link to demonstration screenshots and images:

- https://github.gatech.edu/gt-hit-spring2018/MULTI-SITE-EMR-SURVEILLANCE/tree/master/docs

THANK YOU