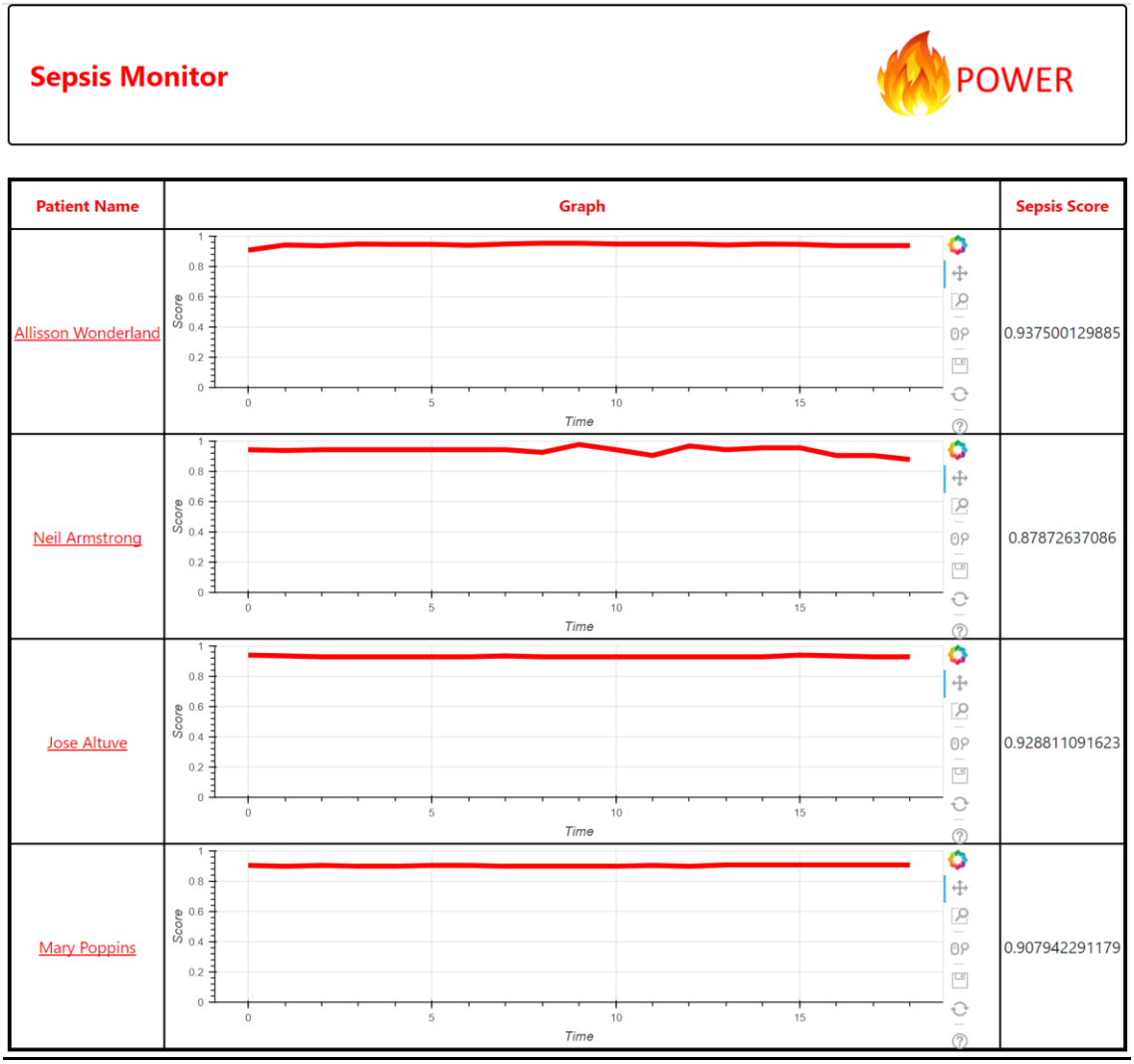
Team Name: FHIR Power
Team Members: Joel Henry, Jeff Mals, Wei Lv, Violette Ogega, Dennis Lynch
Project Name: Streaming-Sepsis-Prediction-System-for-Intensive-Care-Units
Github Link: https://github.gatech.edu/gt-hit-fall2017/Streaming-Sepsis-Prediction-System-for-Intensive-Care-Units
Commit id: 4fb51e289375ed5d4caa7e21728b1e0e805fd23e

Manual

Sepsis Predictor View:



_____The Sepsis Predictor view is used to keep track of the Sepsis score for each of the associated patients. To reach this view, after starting the application (as described in the Special Instructions pdf), navigate to localhost:8000/SepsisPredictor. In this view, the leftmost column is the patient name, the middle column is a plot of the sepsis score over time for the patient, and the rightmost column is the current sepsis score for the patient. Within this view, one can navigate through the time plot of a patient's sepsis score using their mouse - by clicking

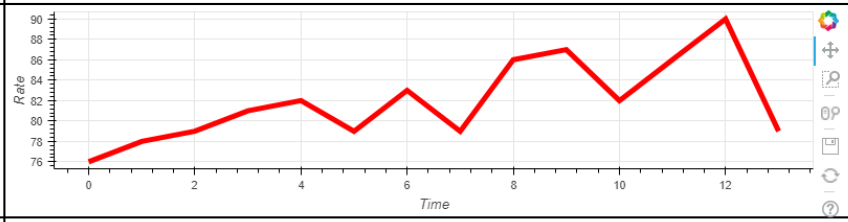
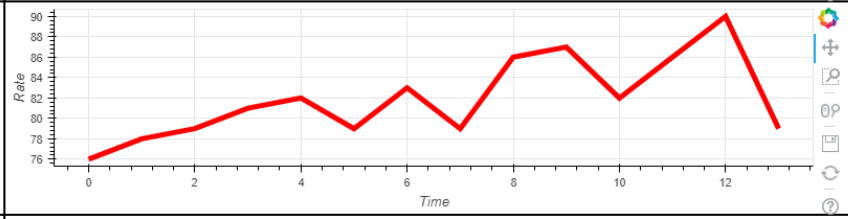
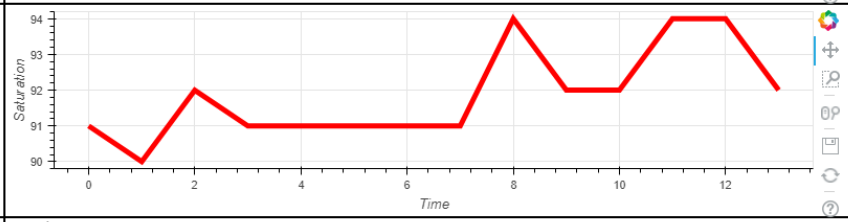
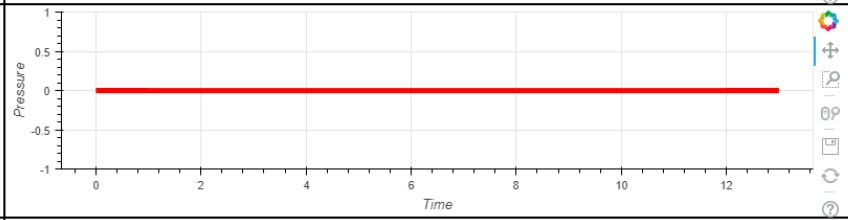
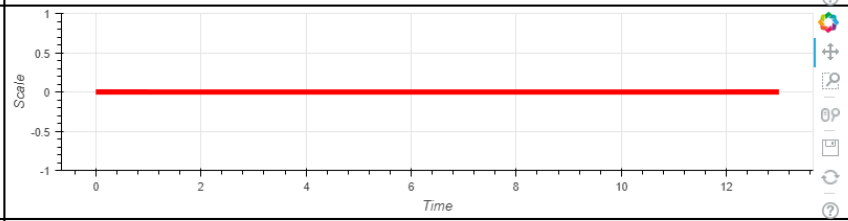
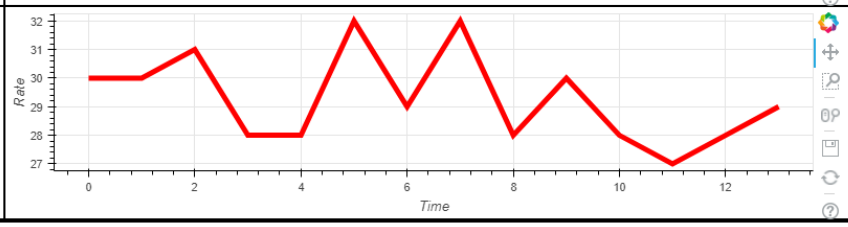
and dragging, the user can view prior time periods, and by scrolling the mouse wheel they can zoom in or out on the plot. The plot is populated automatically, and this view is designed simply to observed data as it enters the FHIR server, and subsequently the local database instance associated with the project.

Patient Detail View:

Sepsis Monitor

POWER

Patient Name: Neil Armstrong Patient ID: 3 Updated: Oct. 2, 2017, 1:54 p.m.

Metric	Graph	Value
Heart Rate		
Temperature		
O2 Saturation		
Blood Pressure		
Glasgow Coma Scale		
Respiratory Rate		

The patient detail view can be found by clicking on the patient name from the Sepsis Predictor view. Within this view, the component data reading names are listed in the leftmost column and the middle column displays the history of those readings over time. This view is

designed to observe the historical data for the six readings that play a role in determining a patient's sepsis score - the graph behaves in the same way as the Sepsis Predictor view graph; the view can be moved by clicking and dragging the mouse, and the scroll wheel can be used to zoom in or out. The data displayed in the graph view is populated automatically as data enters the FHIR server, and subsequently the local database instance. The heading at the top of the page lists the patient's name, the patient's id, and the latest update time for the readings.

Backend FHIR interface:

The backend FHIR interface was designed to add data to the FHIR server, and retrieve data from the server as requested. This interface, located in the github repository at FHIRPower/DataInterface/fhirinterface.py, is utilized as follows:

__init__(self, app_id, api_base):

param app_id: The identifier associated with the application, sent to the FHIR server

param api_base: The base url for the FHIR server being acted upon by the caller

Initializes the SMART on FHIR client instance, to interact with the server

getPatientById(self, patient_id):

param patient_id: The FHIR id associated with the patient being requested

Returns a SMART on FHIR Patient object representing the requested patient resource

getAllPatients(self):

Returns a list of all the patients in the FHIR server instance

getObservationById(self, observation_id):

param observation_id: The FHIR id associated with the observation being requested

Returns a SMART on FHIR Observation object representing the requested observation resource

getAllPatientObservations(self, patient_id):

param patient_id: The FHIR id associated with the patient whose observations are being requested

Return a list of all the observations on the FHIR server associated with the patient

get AllPatientObservationsInRange(self, patient_id, start_time, end_time):

param patient_id: The FHIR id associated with the patient whose observations are being requested

param start_time: The starting timestamp of the time range within which observations should be returned

param end_time: The ending timestamp of the time range within which observations should be returned

Returns a list of all the observations on the FHIR server associated with the patient, which were recorded within a timerange defined by the starting and ending timestamps.

getPatientObservationByLoinc(self, patient_id, loinc_code):

param patient_id: The FHIR id associated with the patient whose observations are being requested

param loinc_code: The LOINC code associated with the observation type being requested

Returns a list of all observations on the FHIR server associated with the patient that falls under the provided LOINC code.

getPatientObservationByTimestamp(self, patient_id, timestamp):

param patient_id: The FHIR id associated with the patient whose observations are being requested

param timestamp: The timestamp for the requested observation, tied to when it was recorded

Returns a list of all observations on the FHIR server associated with the patient that occurs at the specific timestamp.

getPatientObservationsByLoincAndTimestamp(self, patient_id, timestamp, loinc_code):

param patient_id: The FHIR id associated with the patient whose observations are being requested

param timestamp: The timestamp for the requested observation, tied to when it was recorded

param loinc_code: The LOINC code associated with the observation type being requested

Returns an observation that matches the patient id and LOINC code that was recorded at the specified timestamp.

addPatientToServer(self, given_name, family_name):

param given_name: The given (first) name of the patient to be added

param family_name: The family (last) name of the patient to be added. Also known as the patient surname

Adds a new patient resource to the FHIR server, with the provided given and family name added to the patient resource.

addObservationToServer(self, patient_id, value_code, value_unit, value_quantity, coding_code, coding_display, coding_system, timestamp):

param patient_id: The FHIR id associated with the patient whose observations are being requested

param value_code: The datatype code associated with the observation value

param value_unit: The unit associated with the observation value

param value_quantity: The quantity associated with the observation value

param coding_code: The code associated with the observation coding system resource

param coding_display: The display value for the observation coding system resource

param coding_system: The coding system the observation code adheres to

param timestamp: The timestamp associated with the observation record

Creates an observation resource on the FHIR server that contains the provided information for both the observation value, observation time, and the coding system used to code the observation.

removePatientFromServer(self, patient_id):

param patient_id: The FHIR id associated with the patient whose observations are being requested

Removes the associated patient from the FHIR server based on the provided id.

removeObservationFromServer(self, observation_id):

param patient_id: The FHIR id associated with the patient whose observations are being requested

Removes the associated observations from the FHIR server based on the provided id.

getCreatedId(self, input_json):

param input_json: The json provided by the FHIR server upon the creation of a resource

Parses the json provided by the FHIR server, and returns the FHIR id assigned to the FHIR resource

getPagedResults(self, initial_bundle):

param initial_bundle: A FHIR bundle that packages together resources

Pages through the bundled resource, and returns a list of associated SMART on FHIR models based on the contents of the bundle.