Design Document of
Ventilator Weaning Assistant System (VWAS)

Xu Shi, Xiaoyang Xia, Xingdan Wang, Kunming Zhu, Chen Hu, Chan Manfred
xshi307@gatech.edu, xxia44@gatech.edu, xwang3010@gatech.edu,

kzhu81@gatech.edu, chu93@gatech.edu, mchan83@gatech.edu

1 DESIGN CONSIDERATIONS

1.1 Assumptions

The goal of this project is to create a web app that provides ventilator weaning recommendation for patients on ventilators in a real-time basis. Assumptions including:

- Patient data would be generated by Synthea.
- Six body features would be selected to build the model and be utilized in interface design/implementation.
- This web app would display recommendations in two approaches: rule-based model and deep learning model within Tensorflow.
- The web app would have two views:
  - ✧ **Clinician view**: display patients with demographics, vital signs, ventilator status, two recommendations, etc.
  - ✧ **Population view**: display age/gender/race distributions, ventilator availability, counts of days on ventilator, etc.
- The supported public server for the web app would be UHN/HAPI server (R4 FHIR).

1.2 Constraints

- The data generated by Synthea is not real patient data and can be unbalanced. The web app could be used to show the working function and be applied into clinical practice when real patient data is accessible.
- The body features would be selected manually by criteria research. However, ventilator weaning is a very complicated clinical decision based on various observations and conditions of patients. The web app can be used for reference, but the physicians should be the final decision-maker.
- The public server will be regularly purged and reloaded with fixed test data, so it will delete the patient data we uploaded anytime, which triggers challenges in testing and maintenance.

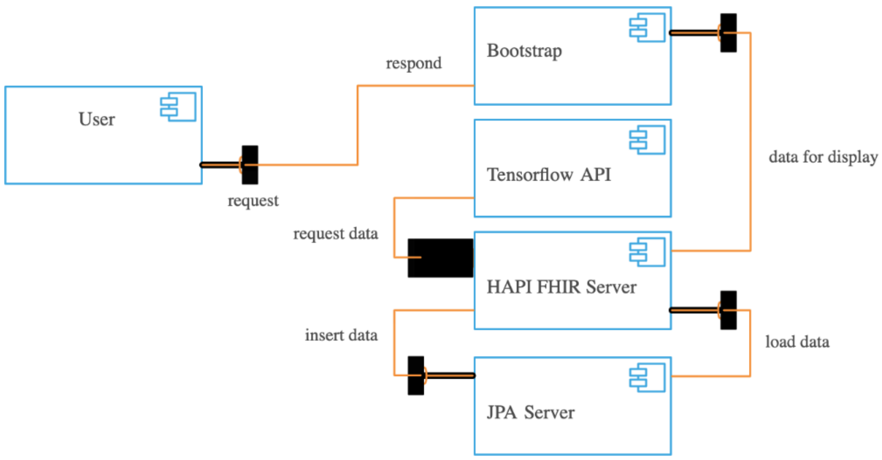
1.3 System Environment

- **Hardware**: desktops, tablets, smartphones.
- **Software**: web browser.

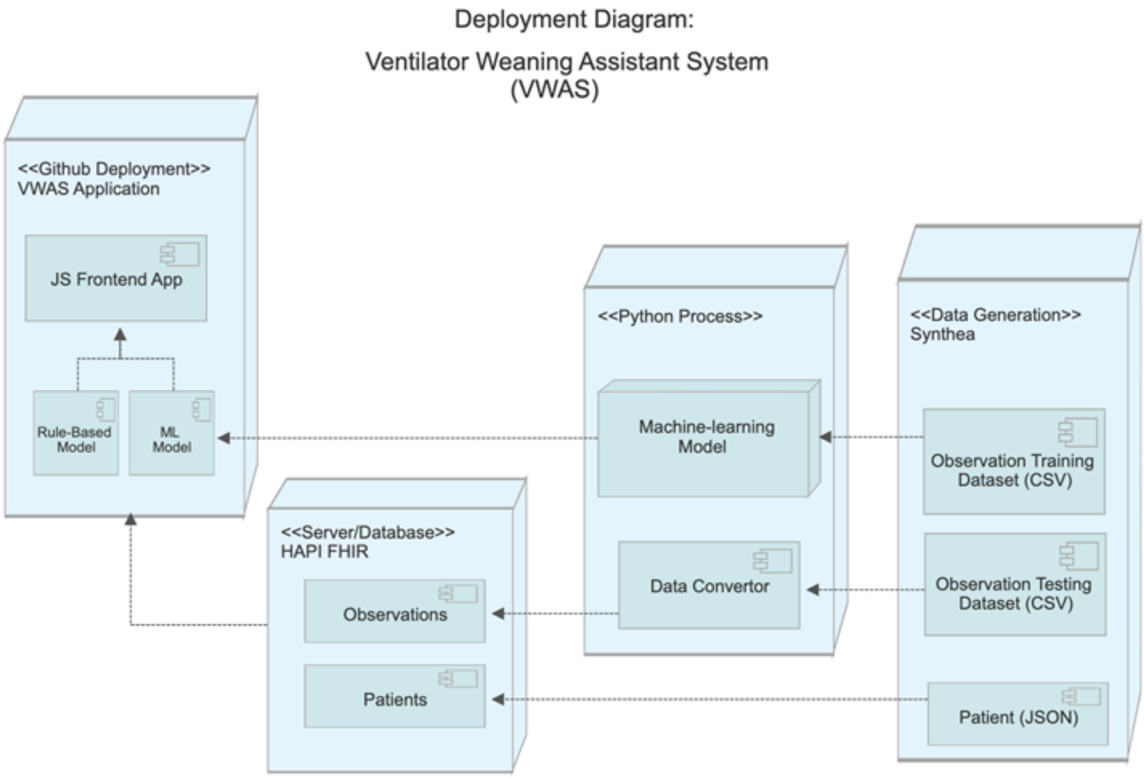
2 ARCHITECTURAL DESIGN

Patient data will be generated by Synthea, and if necessary, processed to fit the required format(s). Generated data will then be pushed to a FHIR server and will be queried based on the user via a web interface. Our web-based app then presents the FHIR resource to the user. The predetermined algorithm and machine learning tools will process the data and present the recommendation in the Clinic View alongside the FHIR resource.

2.1 Component Diagram



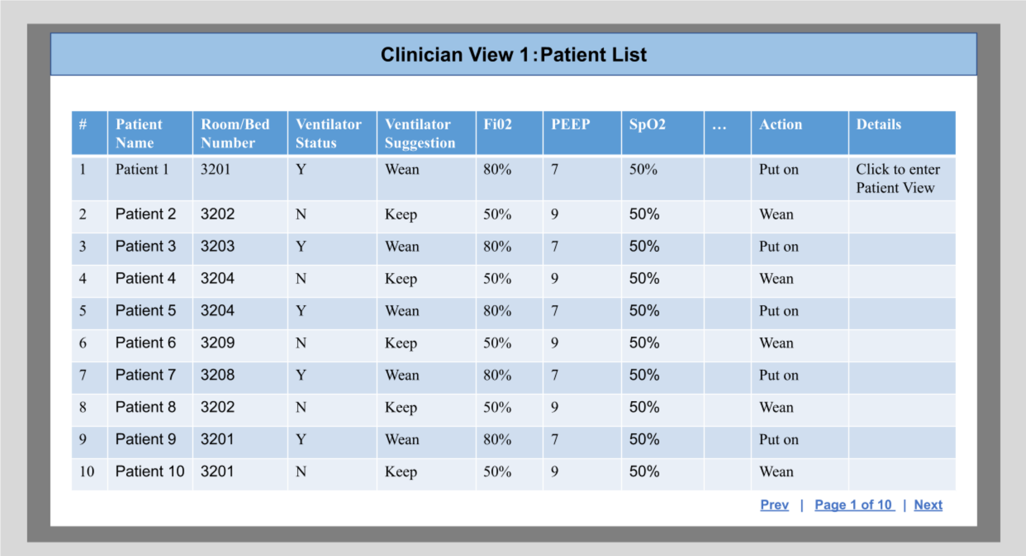
2.2 Deployment Diagram



3 USER INTERFACE DESIGN

We have some preliminary thoughts on the final look of our apps. For the clinician view, it should first have a list of the patients that are on the ventilator and some basic information about each patient. For the screen of each patient, it should have basic personal information, vital signs relevant to the weaning protocols, and also some other critical medical information. Ideally, we would display how these metrics change over time, as well as update the prediction based on the latest data. The ultimate goal is to provide clinical decision support for starting the ventilator weaning process, so that a doctor can be notified, and can start the process as soon as safely possible. Below are a few mock screens based on our current thoughts.

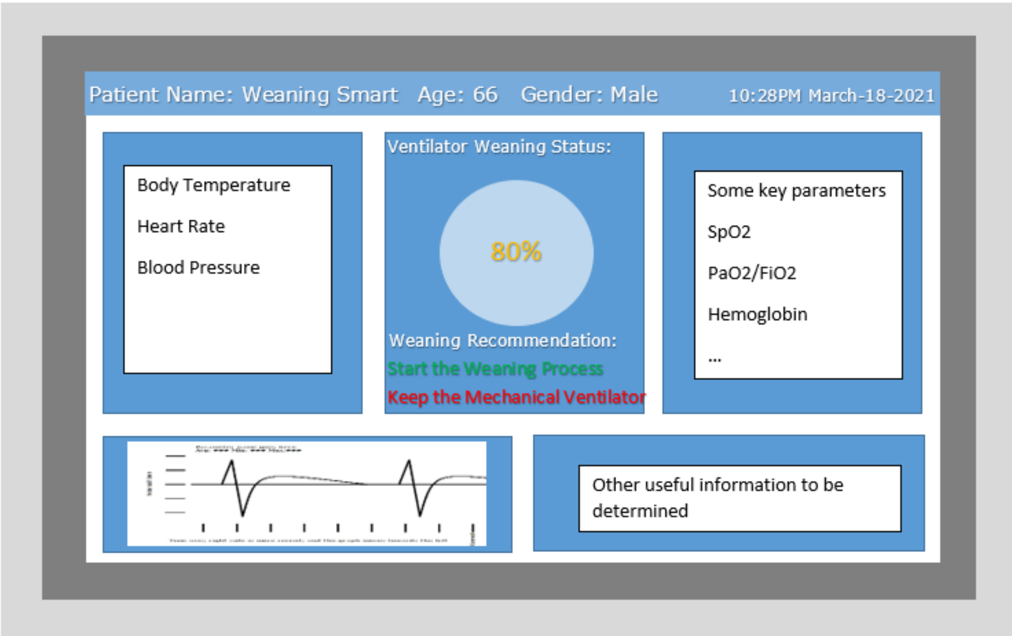
3.1 Clinician view



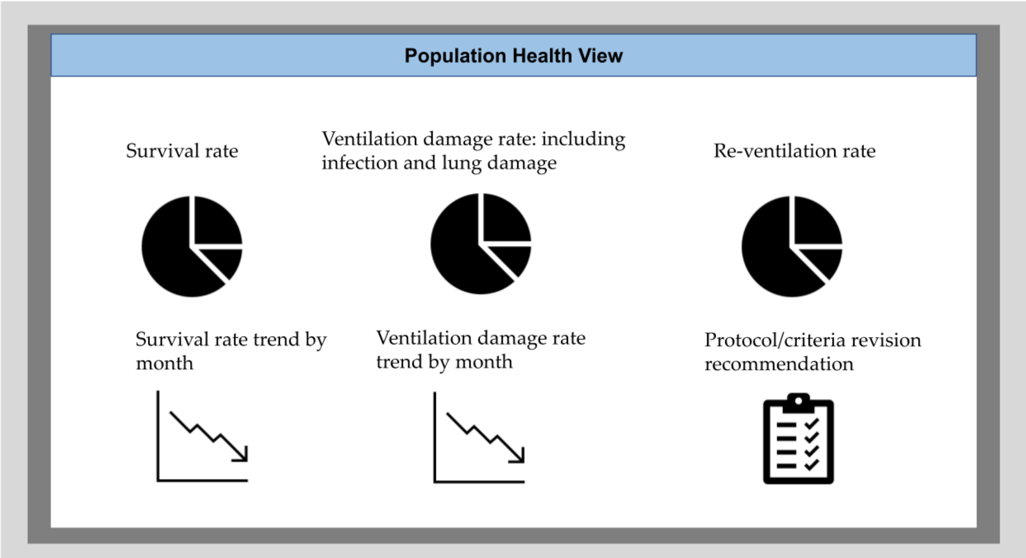
Clinician View 1 : Patient List

#	Patient Name	Room/Bed Number	Ventilator Status	Ventilator Suggestion	Fi02	PEEP	SpO2	...	Action	Details
1	Patient 1	3201	Y	Wean	80%	7	50%		Put on	Click to enter Patient View
2	Patient 2	3202	N	Keep	50%	9	50%		Wean	
3	Patient 3	3203	Y	Wean	80%	7	50%		Put on	
4	Patient 4	3204	N	Keep	50%	9	50%		Wean	
5	Patient 5	3204	Y	Wean	80%	7	50%		Put on	
6	Patient 6	3209	N	Keep	50%	9	50%		Wean	
7	Patient 7	3208	Y	Wean	80%	7	50%		Put on	
8	Patient 8	3202	N	Keep	50%	9	50%		Wean	
9	Patient 9	3201	Y	Wean	80%	7	50%		Put on	
10	Patient 10	3201	N	Keep	50%	9	50%		Wean	

Prev | Page 1 of 10 | Next



3.2 Population view



4 ALGORITHMS DESIGN

In addition, we could use logic functions to determine if the patient still needs to wear the ventilator or not, but we want to do something more than just use a linear function to decide.

There are two further goals which we want to achieve:

First, we want to predict the future condition for the patient.

Second, the most important one, we want to avoid some cases that although the condition of the patient is qualified for taking off the ventilator, the patient could get potential harm by doing so.

4.1 Rule-based algorithm

Rule Based algorithm calculated the six health indicators satisfaction percentage. The six indicators include: heart rate, arterial pH, PEEP respiratory system, arterial O2 saturation, systolic blood pressure, and oxygen/inspired gas setting ventilator (FiO2).

If at least five indicators are satisfied (five green traffic lights), the rule-based algorithm will recommend the medical staff to wean the ventilator from patients.

Else, the rule-based algorithm will recommend keeping the ventilator on the patient.

4.2 Machine learning algorithm

The training data is generated by the software, so it is a toy dataset, but the functions are workable, so we can switch to a real dataset once we can access it.

Because the interface is written by JavaScript, we switch the machine learning framework to Tensorflow.js.

4.2.1 *Parameters of ML Model*

Although we only have six features in our training dataset, we prepare a CNN model with (512, 256, 128, 64) layers for future datasets which may be more complex. The output layer is currently set to three for weaned, death, and on ventilators. The output format could be changed in the future too.

4.2.2 *Result and Future*

Based on our toy dataset, the model can get 95% accuracy on the test dataset. Considering the resources that we have; this result is acceptable.

If we want to use other datasets, we can easily modify the model to fix the input features by scaling or weighting them, and we can also change the output layer to make it suitable for the output classes.

5 REFERENCE

1. Web application | Editable UML Component Diagram Template on Creately. (2021, April 23). Creately. https://creately.com/diagram/example/iyskfw421/Web%20application
2. Deployment Diagram - Web Application. (2021, April 23). Smartdraw. https://www.smartdraw.com/deployment-diagram/examples/deployment-diagram-web-application/