Emergency Department Triage System - Project Deployment

Term: Spring 2020

NewHAT

Project 54

1 TEAM DETAILS

1.1 Team Members/ Team Member Roles & Responsibilities

Nathan Hadder – Project Manager – Responsible for planning and overseeing the project and that they are completed in a timely manner. Project Managers monitor and report progress as well as plan and designate resources.

James Campell – QA - Ensures that the final product is of high quality and has little to no bugs. In charge of implementation of test activities, and the detection and resolution of problems.

William Callan – Developer - Provide ingenuity and development for the software providing the actual coding efforts.

Vaneet Verma – Developer - Provide ingenuity and development for the software providing the actual coding efforts.

Vidya Menon- Developer - Provide ingenuity and development for the software providing the actual coding efforts.

1.2 TA Mentor

Ronnie Jones

1.3 Industry Advisor
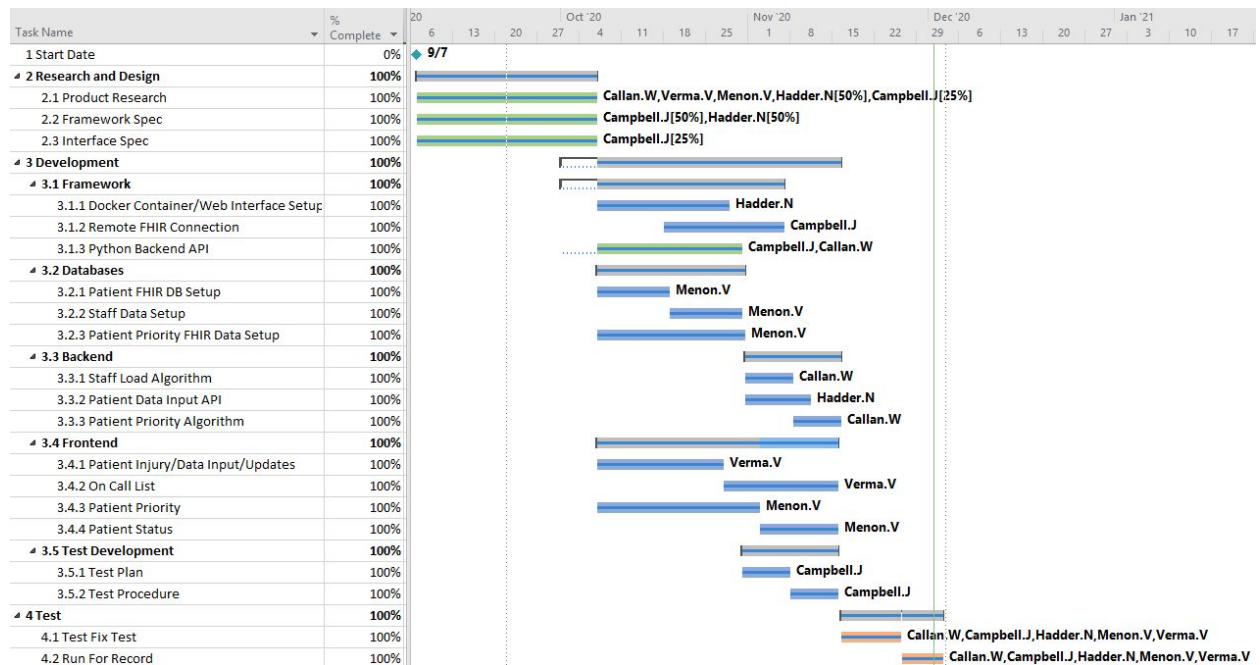
None

2   QUICK DESCRIPTION OF APPLICATION

A hospital's Emergency Department should be prepared in the event of a catastrophic event resulting in mass casualties. This means there should be a Triage process in place to effectively manage ED workers to ensure patients get the care needed in a timely manner. The ability to manage such an event requires processes that include:

- Identify doctors and nurses who are on-call

- Patient evaluation and assessment

- Patient rating to prioritize patients who need immediate care, patients with little survival chances and patients who may need to be transported to external trauma center, and

- Patient progress tracking

The purpose of this application is to help hospital staff so that triage patients can get timely attention before they enter the hospital.

-Triage is the process of classifying patients according to the severity of their injuries.

-Triage system determines how quickly they need care to avoid patients' deaths.

-Many hospitals use triage when medical-care system is overloaded and there are people who need care.

-Hospitals use a 5-level system, considering current resources and the status of the patient (ESI).

-Various codes exist in FHIR to display these levels

3 FINAL GANTT CHART

Task Name	% Complete
1 Start Date	0%
2 Research and Design	100%
2.1 Product Research	100%
2.2 Framework Spec	100%
2.3 Interface Spec	100%
3 Development	100%
3.1 Framework	100%
3.1.1 Docker Container/Web Interface Setup	100%
3.1.2 Remote FHIR Connection	100%
3.1.3 Python Backend API	100%
3.2 Databases	100%
3.2.1 Patient FHIR DB Setup	100%
3.2.2 Staff Data Setup	100%
3.2.3 Patient Priority FHIR Data Setup	100%
3.3 Backend	100%
3.3.1 Staff Load Algorithm	100%
3.3.2 Patient Data Input API	100%
3.3.3 Patient Priority Algorithm	100%
3.4 Frontend	100%
3.4.1 Patient Injury/Data Input/Updates	100%
3.4.2 On Call List	100%
3.4.3 Patient Priority	100%
3.4.4 Patient Status	100%
3.5 Test Development	100%
3.5.1 Test Plan	100%
3.5.2 Test Procedure	100%
4 Test	100%
4.1 Test Fix Test	100%
4.2 Run For Record	100%

3.1   Link

https://github.gatech.edu/gt-cs6440-hit-fall2020/Emergency-Department-Triage-System/blob/master/Final%20Delivery/GanttChart/CS6440%20Gantt%20Chart%20-%20Sprint%208.mpp

https://github.gatech.edu/gt-cs6440-hit-fall2020/Emergency-Department-Triage-System/blob/master/Final%20Delivery/GanttChart/CS6440%20Gantt%20Chart%20-%20Sprint%208.PNG

4   DEPLOYMENT DETAILS

The application is deployed on the HDAP server details are below.

4.1   Dockerfiles

4.1.1   *Frontend*

Description

This container runs our angular frontend application. It exposes the port 4200 for utilization of the application.

Location

https://github.gatech.edu/gt-cs6440-hit-fall2020/Emergency-Department-Triage-System/blob/deploy/app/front-end/Dockerfile

Specifics

Base Image: Node-12

Exposed Port: 4200

4.1.2 Backend

Description

This container runs our flask application for the backend. The application provides an API for the frontend to access information for display and an API for pushing data to both the FHIR server and Postgres database. Port 5000 is exposed for the running API.

Location

https://github.gatech.edu/gt-cs6440-hit-fall2020/Emergency-Department-Triage-System/blob/deploy/app/back-end/Dockerfile

Specifics

Base Image: ubuntu 20.10

Installed Software: python3-pip, libpq-dev, flask

Exposed Port: 5000

Description

This container runs the Postgres database for extra data not able to be stored in on the FHIR server. The port 5432 is exposed for database access.

4.1.3 Database

Location

https://github.gatech.edu/gt-cs6440-hit-fall2020/Emergency-Department-Triage-System/blob/deploy/app/db/Dockerfile

Specifics

Base Image: postgres latest

Exposed Port: 5432

4.2   .drone.yml

4.2.1   *Location*

https://github.gatech.edu/gt-cs6440-hit-fall2020/Emergency-Department-Triage-System/blob/deploy/.drone.yml

4.2.2   *Specifics*

Steps

1. deploy_application_frontend
    a. Repo

        gt-build.hdap.gatech.edu/emergency-department-triage-system
    b. Dockerfile

        app/front-end/Dockerfile
2. Deploy_application_backend
    a. Repo

        gt-build.hdap.gatech.edu/emergency-department-triage-system-backend
    b. Dockerfile

        app/back-end/Dockerfile
3. edts-db
    a. Repo

        gt-build.hdap.gatech.edu/emergency-department-triage-system-db
    b. Dockerfile

        app/db/Dockerfile
4. get_chart_builder

5. copy_namespace
6. deploy_to_k8s

Trigger

Branch: deploy

Event: push

4.3 values.yaml

4.3.1 *Location*

https://github.gatech.edu/gt-cs6440-hit-fall2020/Emergency-Department-Triage-System/blob/deploy/values.yaml

4.3.2 *Specifics*

Namespace

ns-emergency-department-triage-system-2

Frontend

Incoming Path: /emergency-department-triage-system-app

Repository: gt-build.hdap.gatech.edu/emergency-department-triage-system

Port: 4200

Backend

Incoming Path: /emergency-department-triage-system-api

Repository: gt-build.hdap.gatech.edu/emergency-department-triage-system-backend

Port: 4200

Database

Repository: gt-build.hdap.gatech.edu/emergency-department-triage-system-db

Port: 5432

Provide drone configuration information which includes any specific paths, namespace, ports, container info, etc. (this can be on the app readme or a different md file.)

Link to .drone.yaml /.drone.yml files

Link to any other deployment files

4.4   Github Repository

4.4.1   *Link:*

https://github.gatech.edu/gt-cs6440-hit-fall2020/Emergency-Department-Triage-System
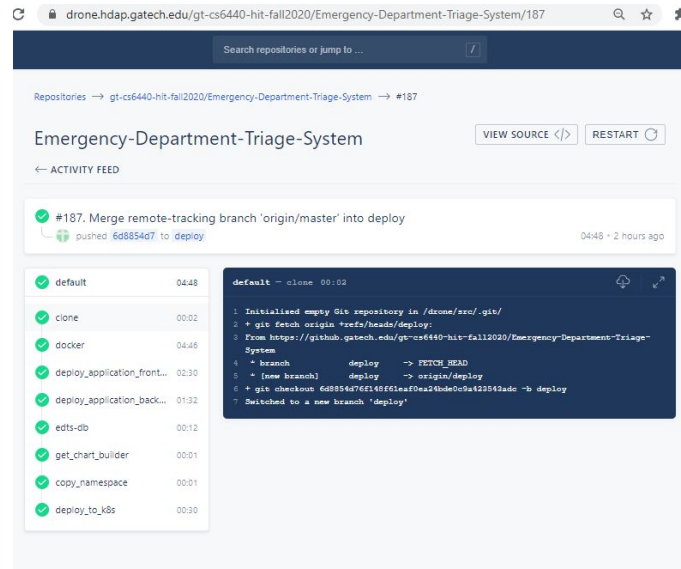
4.4.2   *Branch*

deploy

4.4.3   *Tag*

EDTS_Final_Delivery

4.4.4   *Final Commit ID*

6d8854d76f148f61eaf0ea24bde0c9a423543adc

5   Deployed Application

5.1   Drone Build Logs Screenshot

5.1.1   *Link to final delivery*

https://drone.hdap.gatech.edu/gt-cs6440-hit-fall2020/Emergency-Department-Triage-System/187

5.1.2   *Clone Logs*

Screen Shot
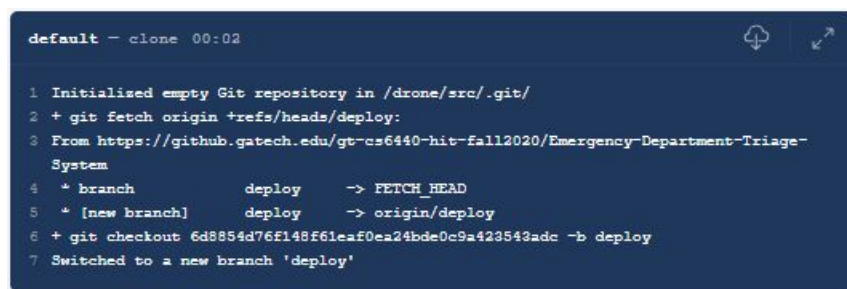
Link

https://github.gatech.edu/gt-cs6440-hit-fall2020/Emergency-Department-Triage-System/blob/master/Final%20Delivery/DroneLogs/logs_gt-cs6440-hit-fall2020_Emergency-Department-Triage-System_1_1.log

5.1.3 Docker logs

Screen Shot


```
default — docker 04:46

1   Generating RSA private key, 4096 bit long modulus (2 primes)
2   ...................................++++
3   ...............................++++
4   e is 65537 (0x010001)
5   Generating RSA private key, 4096 bit long modulus (2 primes)
6   .............................................................++++
7   ....................................................++++
8   e is 65537 (0x010001)
9   Signature ok
10  subject=CN = docker:dind server
11  Getting CA Private Key
12  /certs/server/cert.pem: OK
13  Generating RSA private key, 4096 bit long modulus (2 primes)
14  .................................................++++
15  ..........................................++++
16  e is 65537 (0x010001)
17  Signature ok
18  subject=CN = docker:dind client
19  Getting CA Private Key
20  /certs/client/cert.pem: OK
21  time="2020-12-03T01:17:36.831747921Z" level=info msg="Starting up"
22  time="2020-12-03T01:17:36.835699256Z" level=warning msg="could not change group
    /var/run/docker.sock to docker: group docker not found"
23  time="2020-12-03T01:17:36.837925023Z" level=info msg="libcontainerd: started new
    containerd process" pid=52
24  time="2020-12-03T01:17:36.838181948Z" level=info msg="parsed scheme: \"unix\""
    module=grpc
25  time="2020-12-03T01:17:36.838355326Z" level=info msg="scheme \"unix\" not
    registered, fallback to default scheme" module=grpc
26  time="2020-12-03T01:17:36.838428006Z" level=info msg="ccResolverWrapper: sending
    update to cc: {[{unix:///var/run/docker/containerd/containerd.sock 0  <nil>}]
    <nil>}" module=grpc
27  time="2020-12-03T01:17:36.838599040Z" level=info msg="ClientConn switching
    balancer to \"pick_first\"" module=grpc
28  time="2020-12-03T01:17:36.865503635Z" level=info msg="starting containerd"
    revision=35bd7a5f69c13e1563af8a93431411cd9ecf5021 version=v1.2.12
29  time="2020-12-03T01:17:36.866073210Z" level=info msg="loading plugin
    "io.containerd.content.v1.content"..." type=io.containerd.content.v1
30  time="2020-12-03T01:17:36.866292047Z" level=info msg="loading plugin
    "io.containerd.snapshotter.v1.btrfs"..." type=io.containerd.snapshotter.v1
31  time="2020-12-03T01:17:36.866796702Z" level=warning msg="failed to load plugin
    io.containerd.snapshotter.v1.btrfs" error="path
    /var/lib/docker/containerd/daemon/io.containerd.snapshotter.v1.btrfs must be a
    btrfs filesystem to be used with the btrfs snapshotter"
32  time="2020-12-03T01:17:36.866920072Z" level=info msg="loading plugin
    "io.containerd.snapshotter.v1.aufs"..." type=io.containerd.snapshotter.v1
33  time="2020-12-03T01:17:36.884225197Z" level=warning msg="failed to load plugin
    io.containerd.snapshotter.v1.aufs" error="modprobe aufs failed: "ip: can't find
    device 'aufs'\naufs                241664  0 \nmodprobe: can't change directory
    to '/lib/modules': No such file or directory\n": exit status 1"
34  time="2020-12-03T01:17:36.884240607Z" level=info msg="loading plugin
```

Link

https://github.gatech.edu/gt-cs6440-hit-fall2020/Emergency-Department-Triage-System/blob/master/Final%20Delivery/DroneLogs/logs_gt-cs6440-hit-fall2020_Emergency-Department-Triage-System_1_2.log

5.1.4   *Deploy Application Frontend*

Screen Shot

```
default — deploy_application_frontend 02:30

1     latest: Pulling from plugins/docker
2     Digest: sha256:60f36379eb3f1b27a61f7c0a275dd260387a00af1144345976475b2a35e5ee4f
3     Status: Image is up to date for plugins/docker:latest
4     + /usr/local/bin/dockerd --data-root /var/lib/docker --
      host=unix:///var/run/docker.sock
5     Registry credentials or Docker config not provided. Guest mode enabled.
6     + /usr/local/bin/docker version
7     Client: Docker Engine - Community
8      Version:            19.03.8
9      API version:        1.40
10     Go version:         go1.12.17
11     Git commit:         afacb8b7f0
12     Built:              Wed Mar 11 01:22:56 2020
13     OS/Arch:            linux/amd64
14     Experimental:       false
15
16    Server: Docker Engine - Community
17     Engine:
18      Version:           19.03.8
19      API version:       1.40 (minimum version 1.12)
20      Go version:        go1.12.17
21      Git commit:        afacb8b7f0
22      Built:             Wed Mar 11 01:30:32 2020
23      OS/Arch:           linux/amd64
24      Experimental:      false
25     containerd:
26      Version:           v1.2.13
27      GitCommit:         7ad184331fa3e55e52b890ea95e65ba581ae3429
28     runc:
29      Version:           1.0.0-rc10
30      GitCommit:         dc9208a3303feef5b3839f4323d9beb36df0a9dd
31     docker-init:
32      Version:           0.18.0
33      GitCommit:         fec3683
34    + /usr/local/bin/docker info
35    Client:
```

Link

https://github.gatech.edu/gt-cs6440-hit-fall2020/Emergency-Department-Triage-System/blob/master/Final%20Delivery/DroneLogs/logs_gt-cs6440-hit-fall2020_Emergency-Department-Triage-System_1_3.log

5.1.5 Deploy Application Backend

Screen Shot

```
default — deploy_application_backend  01:32
1    latest: Pulling from plugins/docker
2    Digest: sha256:60f36379eb3f1b27a61f7c0a275dd260387a00af1144345976475b2a35e5ee4f
3    Status: Image is up to date for plugins/docker:latest
4    + /usr/local/bin/dockerd --data-root /var/lib/docker --
     host=unix:///var/run/docker.sock
5    Registry credentials or Docker config not provided. Guest mode enabled.
6    + /usr/local/bin/docker version
7    Client: Docker Engine - Community
8     Version:           19.03.8
9     API version:       1.40
10     Go version:       go1.12.17
11     Git commit:       afacb8b7f0
12     Built:            Wed Mar 11 01:22:56 2020
13     OS/Arch:          linux/amd64
14     Experimental:     false
15
16    Server: Docker Engine - Community
17     Engine:
18      Version:          19.03.8
19      API version:      1.40 (minimum version 1.12)
20      Go version:       go1.12.17
21      Git commit:       afacb8b7f0
22      Built:            Wed Mar 11 01:30:32 2020
23      OS/Arch:          linux/amd64
24      Experimental:     false
25     containerd:
26      Version:          v1.2.13
27      GitCommit:        7ad184331fa3e55e52b890ea95e65ba581ae3429
28     runc:
29      Version:          1.0.0-rc10
30      GitCommit:        dc9208a3303feef5b3839f4323d9beb36df0a9dd
31     docker-init:
32      Version:          0.18.0
33      GitCommit:        fec3683
34    + /usr/local/bin/docker info
35    Client:
```

Link

https://github.gatech.edu/gt-cs6440-hit-fall2020/Emergency-Department-Triage-System/blob/master/Final%20Delivery/DroneLogs/logs_gt-cs6440-hit-fall2020_Emergency-Department-Triage-System_1_4.log

5.1.6   Edts-db

Screen Shot

```
default — edts-db 00:12                                          ⌄    ⤢

1     latest: Pulling from plugins/docker
2     Digest: sha256:60f36379eb3f1b27a61f7c0a275dd260387a00af1144345976475b2a35e5ee4f
3     Status: Image is up to date for plugins/docker:latest
4     + /usr/local/bin/dockerd --data-root /var/lib/docker --
      host=unix:///var/run/docker.sock
5     Registry credentials or Docker config not provided. Guest mode enabled.
6     + /usr/local/bin/docker version
7     Client: Docker Engine - Community
8      Version:            19.03.8
9      API version:        1.40
10     Go version:         go1.12.17
11     Git commit:         afacb8b7f0
12     Built:              Wed Mar 11 01:22:56 2020
13     OS/Arch:            linux/amd64
14     Experimental:       false
15
16    Server: Docker Engine - Community
17     Engine:
18      Version:           19.03.8
19      API version:       1.40 (minimum version 1.12)
20      Go version:        go1.12.17
21      Git commit:        afacb8b7f0
22      Built:             Wed Mar 11 01:30:32 2020
23      OS/Arch:           linux/amd64
24      Experimental:      false
25     containerd:
26      Version:           v1.2.13
27      GitCommit:         7ad184331fa3e55e52b890ea95e65ba581ae3429
28     runc:
29      Version:           1.0.0-rc10
30      GitCommit:         dc9208a3303feef5b3839f4323d9beb36df0a9dd
31     docker-init:
32      Version:           0.18.0
33      GitCommit:         fec3683
34    + /usr/local/bin/docker info
35    Client:
```

Link

https://github.gatech.edu/gt-cs6440-hit-fall2020/Emergency-Department-Triage-System/blob/master/Final%20Delivery/DroneLogs/logs_gt-cs6440-hit-fall2020_Emergency-Department-Triage-System_1_5.log

5.1.7 Get_chart_builder

Screen Shot


```
default — get_chart_builder 00:01

1 + git clone -b cs6440 https://github.gatech.edu/es130/HDAP_helm_deploy_templates.git
2 Cloning into 'HDAP_helm_deploy_templates'...
```

Link

https://github.gatech.edu/gt-cs6440-hit-fall2020/Emergency-Department-Triage-System/blob/master/Final%20Delivery/DroneLogs/logs_gt-cs6440-hit-fall2020_Emergency-Department-Triage-System_1_6.log

5.1.8 Copy_namespace

Screen Shot


```
default — copy_namespace 00:01

1 + userns=ns-emergency-department-triage-system-2
2 + sed -i "s/REPLACE_WITH_NAMESPACE/$userns/g"
  ./HDAP_helm_deploy_templates/deploy/Dockerfile
```

Link

https://github.gatech.edu/gt-cs6440-hit-fall2020/Emergency-Department-Triage-System/blob/master/Final%20Delivery/DroneLogs/logs_gt-cs6440-hit-fall2020_Emergency-Department-Triage-System_1_7.log

5.1.9 Deploy_to_k8s

Screen Shot

```
default — deploy_to_k8s 00:30

1    + sleep 10
2    + docker build -t deploytok8s -f ./HDAP_helm_deploy_templates/deploy/Dockerfile .
3    Sending build context to Docker daemon    16MB

4    Step 1/6 : FROM gt-build.hdap.gatech.edu/hdap_helm_prj_chart_builder:1.0.0 AS
     templateBuilder
5    1.0.0: Pulling from hdap_helm_prj_chart_builder
6    2004ee2154b4: Pulling fs layer
7    3a3f1549d492: Pulling fs layer
8    84d57fc99c15: Pulling fs layer
9    9ce8c702093d: Pulling fs layer
10   907937f57d50: Pulling fs layer
11   9ce8c702093d: Waiting
12   db328c19241d: Pulling fs layer
13   907937f57d50: Waiting
14   162753355b97: Pulling fs layer
15   3068faec39a1: Pulling fs layer
16   c31994f0e369: Pulling fs layer
17   142f8c8c4bb4: Pulling fs layer
18   305f58b5a55f: Pulling fs layer
19   ebafbb26c704: Pulling fs layer
20   94b7a90a293e: Pulling fs layer
21   db328c19241d: Waiting
22   162753355b97: Waiting
23   3068faec39a1: Waiting
24   c31994f0e369: Waiting
25   305f58b5a55f: Waiting
26   94b7a90a293e: Waiting
27   142f8c8c4bb4: Waiting
28   ebafbb26c704: Waiting
29   84d57fc99c15: Verifying Checksum
30   84d57fc99c15: Download complete
31   9ce8c702093d: Verifying Checksum
32   9ce8c702093d: Download complete
33   907937f57d50: Verifying Checksum
34   907937f57d50: Download complete
```
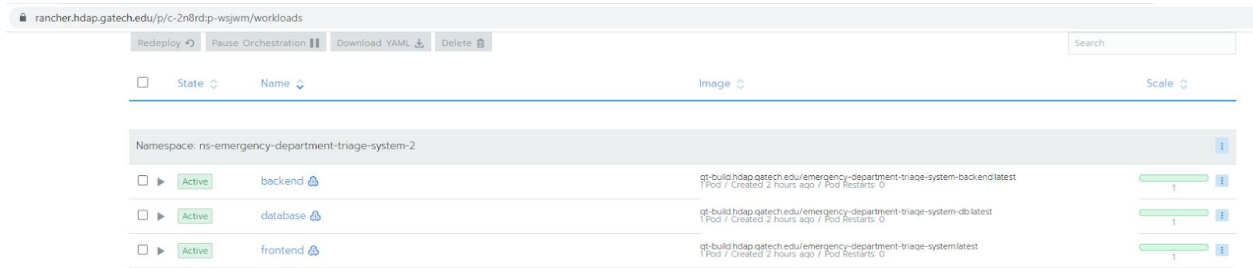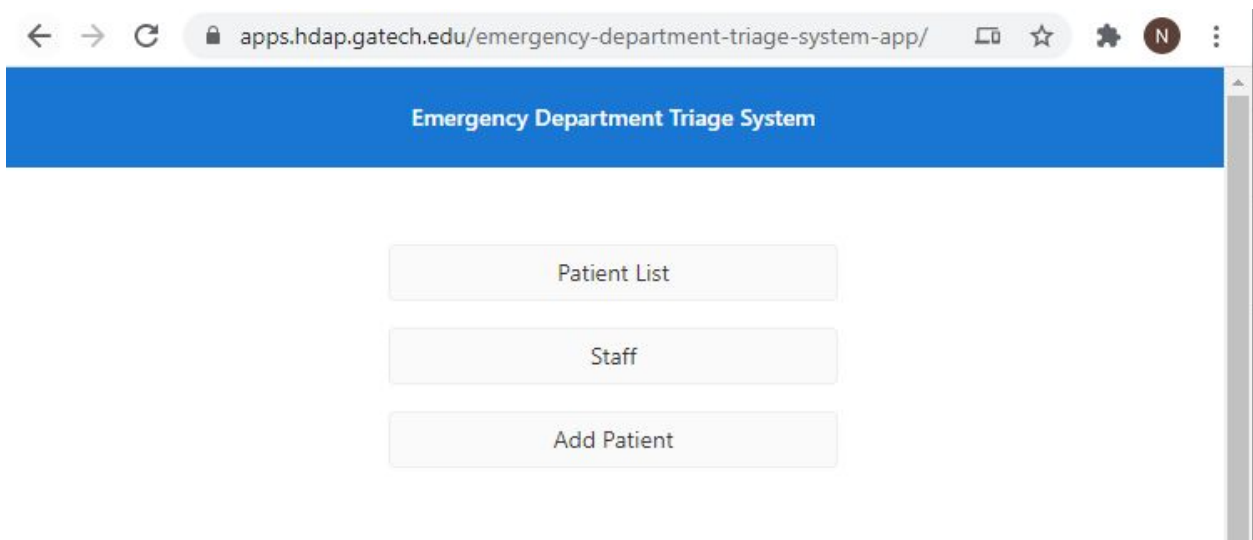
Link

https://github.gatech.edu/gt-cs6440-hit-fall2020/Emergency-Department-Triage-System/blob/master/Final%20Delivery/DroneLogs/logs_gt-cs6440-hit-fall2020_Emergency-Department-Triage-System_1_8.log

5.2 Rancher Container Running Screenshot

5.3 Application Screenshot

5.4 Steps to run app

Full instructions are in the Application Manual found here:

https://github.gatech.edu/gt-cs6440-hit-fall2020/Emergency-Department-Triage-System/blob/master/Final%20Delivery/Documentation/Application%20Manual.pdf

For a Quick Start follow the following.

5.4.1 *On the server*

The application is deployed remotely on the HDAP server at:

https://apps.hdap.gatech.edu/emergency-department-triage-system-app/

which is the full working application frontend. This application communicates with our backend and database to display the overall program to the user. As mentioned previously, before using the application, please visit:

https://apps.hdap.gatech.edu/emergency-department-triage-system-api/generate

first to generate data for the demonstration. The URL above is our backend and will generate the required information even if it was previously wiped by the FHIR server.

5.4.2 *Locally*

Local deployment of the application is possible if there is some issue with the already-deployed version. To deploy each of the components, first clone the project at https://github.gatech.edu/gt-cs6440-hit-fall2020/Emergency-Department-Triage-System

and then navigate to the root of the project directory. Edit the .localenv file and set the DB_HOST variable to the IP address of your docker host (keep it the same if you are running it on localhost). Finally, run the following scripts in order in separate terminals to start the application:

./build.sh

./startDB.sh

./startFrontend.sh

./startAPI.sh

If you can not run shell scripts on your machine, open each of the files to get the docker commands and directories needed to run the containers.

6   LOGIN INFORMATION FOR APP.

There is currently no login for the Emergency Department Triage System.

7   SUBMISSION TO EXTERNAL MENTOR

There was no external mentor for this project.