

Project Catalog - EMS Patient Data

Team Name: FHIR I/O

Project Name: EMS Patient Data

Team Members:

1. Bryce Montano - bmontano6
2. Juan San Emeterio - jjse3
3. Haixia Shang - hshang7
4. Yu Wang - ywang3134
5. Smitha George - sgeorge65

GitHub link: <https://github.gatech.edu/gt-hit-fall2017/EMS-Patient-Data>

Table of Contents

- 1.0 - Final Delivery
 - 1.1 - Research/Project_research.pdf
 - 1.2 - media
 - 1.3 - Manual - FHIR IO.pdf
 - 1.4 - Manual - FHIR IO.md
 - 1.5 - Final Gantt Chart - FHIR IO.pdf
 - 1.6 - Special Instructions - FHIR IO.md
 - 1.7 - Special Instructions - FHIR IO.pdf
 - 1.8 - catalog.md
 - 1.9 - catalog.pdf
 - 1.10 - Final Presentation.pdf
- 2.0 - fhir-ems-service
 - 2.1 - src
 - 2.1.1 - main
 - 2.1.1.1 - java/com/fhirio/fhiremsservice
 - 2.1.1.1.1 - controller
 - 2.1.1.1.1.1 - AuthenticationController.java
 - 2.1.1.1.1.2 - EmergencyController.java
 - 2.1.1.1.1.3 - FhirController.java
 - 2.1.1.1.1.4 - OrganizationController.java
 - 2.1.1.1.1.5 - PatientController.java
 - 2.1.1.1.1.6 - UserController.java
 - 2.1.1.1.2 - domain
 - 2.1.1.1.2.1 - Address.java
 - 2.1.1.1.2.2 - Authentication.java
 - 2.1.1.1.2.3 - Condition.java
 - 2.1.1.1.2.4 - Conditions.java
 - 2.1.1.1.2.5 - Emergency.java

- 2.1.1.1.2.6 - EmergencyState.java
- 2.1.1.1.2.7 - Measurement.java
- 2.1.1.1.2.8 - Medication.java
- 2.1.1.1.2.9 - Medications.java
- 2.1.1.1.2.10 - Note.java
- 2.1.1.1.2.11 - Organization.java
- 2.1.1.1.2.12 - Patient.java
- 2.1.1.1.2.13 - Token.java
- 2.1.1.1.2.14 - User.java
- 2.1.1.1.3 - service
 - 2.1.1.1.3.1 - AuthenticationService.java
 - 2.1.1.1.3.2 - EmergencyService.java
 - 2.1.1.1.3.3 - OrganizationService.java
 - 2.1.1.1.3.4 - PatientService.java
 - 2.1.1.1.3.5 - UserService.java
 - 2.1.1.1.3.6 - UuidService.java
- 2.1.1.1.4 - FhirClient.java
- 2.1.1.1.5 - FhirEmsServiceApplication.java
- 2.1.1.2 - resources
 - 2.1.1.2.1 - application.properties
 - 2.1.1.2.2 - static
 - 2.1.1.2.2.1 - css
 - 2.1.1.2.2.1.1 - login.css
 - 2.1.1.2.2.2 - js
 - 2.1.1.2.2.2.1 - NewOrganizationalUser.js
 - 2.1.1.2.2.2.2 - dashboard.js
 - 2.1.1.2.2.2.3 - emergency.js
 - 2.1.1.2.2.2.4 - login.js
 - 2.1.1.2.2.2.5 - newemergency.js
 - 2.1.1.2.2.2.6 - newuser.js
 - 2.1.1.2.2.2.7 - patientdetails.js
 - 2.1.1.2.2.3 - dashboard.html
 - 2.1.1.2.2.4 - emergency.html
 - 2.1.1.2.2.5 - index.html
 - 2.1.1.2.2.6 - newOrganizationalUser.html
 - 2.1.1.2.2.7 - newemergency.html
 - 2.1.1.2.2.8 - newuser.html
 - 2.1.1.2.2.9 - patientdetails.html
- 2.1.2 - test/java/com/fhirio/fhiremsservice
 - 2.1.2.1 - FhirEmsServiceApplicationTests.java
 - 2.1.2.2 - service
 - 2.1.2.2.1 - AuthenticationServiceTest.java
 - 2.1.2.2.2 - EmergencyServiceTest.java
 - 2.1.2.2.3 - OrganizationServiceTest.java
 - 2.1.2.2.4 - PatientServiceTest.java
 - 2.1.2.2.5 - PatientUpdateTest.java

- [2.1.2.2.6 - UserServiceTest.java](#)

- [2.2 - .gitignore](#)
- [2.3 - Dockerfile](#)
- [2.4 - pom.xml](#)
- [3.0 - docker-compose.yml](#)

▸ **1.0 - Final Delivery**

This is the top-level directory that contains all our documentation for Deliverables 4/5.

▸ **1.1 - Research/Project_research.pdf**

This file represents all the research that was done in preparation for the development of the application.

▸ **1.2 - media**

This directory contains all the images used in our documentation.

▸ **1.3 - Manual - FHIR IO.pdf**

This is the PDF version of our manual, which contains instructions on how to make use of all the features our application was built to support.

▸ **1.4 - Manual - FHIR IO.md**

This is the markdown version of our manual, which contains instructions on how to make use of all the features our application was built to support.

▸ **1.5 - Final Gantt Chart - FHIR IO.pdf**

This is the final Gantt chart, which includes details about how the work was completed throughout the semester and which team members presented in each deliverable.

▸ **1.6 - Special Instructions - FHIR IO.md**

This document describes how a user could run our application using docker-compose in markdown format.

▸ **1.7 - Special Instructions - FHIR IO.pdf**

This document describes how a user could run our application using docker-compose in pdf format.

▸ **1.8 - catalog.md**

This document describes the contents of the entire repository on a file-by-file basis in markdown format.

▷ **1.9 - catalog.pdf**

This document describes the contents of the entire repository on a file-by-file basis in pdf format.

▷ **1.10 - Final Presentation.pdf**

This is a pdf version of our PowerPoint deck as seen in deliverable 4.

▷ **2.0 - fhir-ems-service**

This is the top-level directory that holds all the source code, dependency management, git files.

▷ **2.1 - src**

This directory holds all of the source code including Java backend service code, front-end code, and unit tests.

▷ **2.1.1 - main**

This directory contains the Java backend service code and front-end code (html/js/css).

▷ **2.1.1.1 - java/com/fhirio/fhiremsservice**

This directory contains the Java backend service code.

▷ **2.1.1.1.1 - controller**

This directory holds all the REST controllers that expose our services.

▷ **2.1.1.1.1.1 - AuthenticationController.java**

This is a REST controller used to expose our AuthenticationService in a RESTful fashion.

▷ **2.1.1.1.1.2 - EmergencyController.java**

This is a REST controller used to expose our EmergencyService in a RESTful fashion.

▷ **2.1.1.1.1.3 - FhirController.java**

This is a REST controller used to expose our FhirClient in a RESTful fashion.

▷ **2.1.1.1.1.4 - OrganizationController.java**

This is a REST controller used to expose our OrganizationService in a RESTful fashion.

‣ **2.1.1.1.5 - PatientController.java**

This is a REST controller used to expose our PatientService in a RESTful fashion.

‣ **2.1.1.1.6 - UserController.java**

This is a REST controller used to expose our UserService in a RESTful fashion.

‣ **2.1.1.1.2 - domain**

This directory holds all the POJOs used by our application to transfer data across the network and within our application.

‣ **2.1.1.1.2.1 - Address.java**

This is a POJO that holds onto a Patient's address information including street address, city, state, and zip code.

‣ **2.1.1.1.2.2 - Authentication.java**

This is a POJO that holds a User's credentials and is used primarily by the AuthenticationService to verify a user's authenticity.

‣ **2.1.1.1.2.3 - Condition.java**

This is a POJO that holds a patient's chronic illness information, includes clinical status, verification status, name, system, code, onset date, ect.

‣ **2.1.1.1.2.4 - Conditions.java**

This is a POJO that holds a list of patient conditions.

‣ **2.1.1.1.2.5 - Emergency.java**

This is a POJO that holds all the pertinent information to a specific emergency that may be pending, active or closed.

‣ **2.1.1.1.2.6 - EmergencyState.java**

This is a POJO that holds possible status of an emergency, including pending, closed and active.

‣ **2.1.1.1.2.7 - Measurement.java**

This is a POJO that holds measurement of a patient including measurement Uuid, name, system, value code, description,value.

▷ **2.1.1.1.2.8 - Medication.java**

This is a POJO that holds the medication information related to a patient involved in an emergency.

▷ **2.1.1.1.2.9 - Medications.java**

This is a POJO that holds a list of patient medications.

▷ **2.1.1.1.2.11 - Organization.java**

This is a POJO that binds together users of the EMS service and Emergencies.

▷ **2.1.1.1.2.12 - Patient.java**

This is a POJO that hold all the health information related to a patient involved in an emergency, including patient profile, address, and patient notes.

▷ **2.1.1.1.2.13 - Token.java**

This is a POJO that system uses to assign token for patient and organization.

▷ **2.1.1.1.2.14 - User.java**

This ia a POJO that holds specific user of the application that may be emergency personnel or call center personnel.

▷ **2.1.1.1.3 - service**

This directory contains the Java backend service code.

▷ **2.1.1.1.3.1 - AuthenticationService.java**

This is a POJO that holds users's anthemtication, and is used to verify the user by username and password.

▷ **2.1.1.1.3.2 - EmergencyService.java**

This is a POJO that initializes the emergency service, gets an emergency by Uuid, saves an emergency pbject to memory, assigns a Uuid to it, and identify patient.

▷ **2.1.1.1.3.3 - OrganizationService.java**

This is a POJO that constructs an organization service, gets an organization by its Uuid, saves/updates/creates an organization,adds organization user, and add emergency to the organization.

▷ **2.1.1.1.3.4 - PatientService.java**

This is a POJO that get patient by Uuid, name, and get/update information of patients, including patient's condition, medication, patient notes.

▸ **2.1.1.1.3.5 - UserService.java**

This is a POJO that provides service related to a user, including create new super user, new organizational user, save user, etc.

▸ **2.1.1.1.3.6 - UuidService.java**

This is a POJO that randomly generate Uuid by the system and assign to an object when is required.

▸ **2.1.1.1.4 - FhirClient.java**

This is a POJO that provides client service for the application, including getting patient list from the server, getting detailed medical information of a patient from the server.

▸ **2.1.1.1.5 - FhirEmsServiceApplication.java**

This is the main class of the application. Run this file will set up the application.

▸ **2.1.1.2 - resources**

This directory contains the front-end code (html/js/css) and application.properties file, which is used to configure aspects of the Java application.

▸ **2.1.1.2.1 - application.properties**

This is properties file for the application.

▸ **2.1.1.2.2 - static**

This directory contains all the html/js/css code.

▸ **2.1.1.2.2.1 - css**

This directory contains the css file for the web pages.

▸ **2.1.1.2.2.1.1 - login.css**

This is the css format file for log in page.

▸ **2.1.1.2.2.2 - js**

This directory contains all javascript files for the HTML web pages.

▷ **2.1.1.2.2.2.1 - NewOrganizationalUser.js**

This is the javascript file for creating a new organizational user by a super user. New user's info will be post to the backend database.

▷ **2.1.1.2.2.2.2 - dashboard.js**

This is the javascript file for the emergency dashboard. Emergencies with different status(pending/active/closed) will ge get from the backend.

▷ **2.1.1.2.2.2.3 - emergency.js**

This is the javascript file to get details of an emergency from the backend.

▷ **2.1.1.2.2.2.4 - login.js**

This is the javascript file to authenticate user to login.

▷ **2.1.1.2.2.2.5 - newemergency.js**

This is the javascript file for creating a new emergency, and post it to the backend .

▷ **2.1.1.2.2.2.6 - newuser.js**

This is the javascript file for creating a new super user.

▷ **2.1.1.2.2.2.7 - patientdetails.js**

This is the javascript file for collecting and showing detailed information of a patient, including profile, medications,etc.

▷ **2.1.1.2.2.3 - dashboard.html**

This is the webpage for viewing emergencies in a organization. The details of each emergency can be shown, and the status of an emergency can be updated by user.

▷ **2.1.1.2.2.4 - emergency.html**

This is the webpage contains details of an emergency and possible patients involved in the emergency.

▷ **2.1.1.2.2.5 - index.html**

This is the webpage for user to log in, or new user to register.

▷ **2.1.1.2.2.6 - newOrganizationalUser.html**

This is the webpage for super user to enroll an new organizationl user. Username and password are required to be filled in.

▷ **2.1.1.2.2.7 - newemergency.html**

This is the webpage for user to fill information of a new emergency. Such as patient name and picking up location.

▷ **2.1.1.2.2.8 - newuser.html**

This is the webpage for user to fill information to register. Such as user's name, password, organization name.

▷ **2.1.1.2.2.9 - patientdetails.html**

This is the webpage showing detailed information of the patient. User can also add patient's note from this page.

▷ **2.1.2 - test/java/com/fhirio/fhiremsservice**

This directory contains all the testing files for the application.

▷ **2.1.2.1 - FhirEmsServiceApplicationTests.java**

▷ **2.1.2.2 - service**

This directory contains the testing files for the services.

▷ **2.1.2.2.1 - AuthenticationServiceTest.java**

This is the testing file to test if a user's authentication is valid, including test on username and passowrd.

▷ **2.1.2.2.2 - EmergencyServiceTest.java**

This is the testing file to test if a emergency's Uuid is valid, giing a set of emergencies.

▷ **2.1.2.2.3 - OrganizationServiceTest.java**

This is the testing file to test if an organization is exitted by checking its Uuid.

▷ **2.1.2.2.4 - PatientServiceTest.java**

This is the testing file to test if a patient is available, giving patient's Uuid and name.

▷ **2.1.2.2.5 - PatientUpdateTest.java**

This is the testing file to test if patient's name, medications, conditions can be get, and if new patient can be added to the database.

▸ **2.1.2.2.6 - UserServiceTest.java**

This is the testing file to test if the user service can get the user by user's Uuid.

▸ **2.2 - .gitignore**

This is the file for Git to determine which files and directories to ignore.

▸ **2.3 - Dockerfile**

This is the docker directory that holds required docker file for the application.

▸ **2.4 - pom.xml**

This is the XML file that contains information about the project and configuration details used by Maven to build the project.

▸ **3.0 - docker-compose.yml**

This is the docker file that used to build and run docker for the application.